

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-306849

(43)Date of publication of application : 21.11.1995

(51)Int.Cl.

G06F 17/14

(21)Application number : 06-167741

(71)Applicant : FUJITSU LTD

(22)Date of filing : 20.07.1994

(72)Inventor : NAKAGAWA AKIRA
KAWAKATSU YASUHIRO
MORIMATSU EIJI
MATSDA KIICHI

(30)Priority

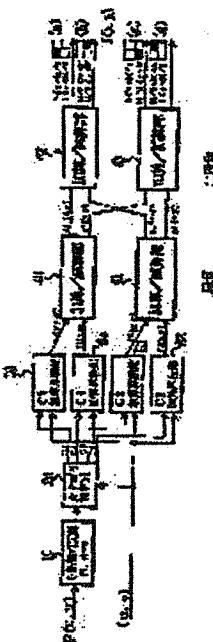
Priority number : 06 74121 Priority date : 18.03.1994 Priority country : JP

(54) TWO-DIMENSIONAL INVERSE DISCRETE COSINE TRANSFORMATION SYSTEM

(57)Abstract:

PURPOSE: To provide the two-dimensional inverse discrete cosine transformation system which can increase the speed of an encoding process with small memory capacity.

CONSTITUTION: According to whether the horizontal and vertical frequencies of a base function that coefficients which are not 0 among DCT coefficients of one object block for the two-dimensional DCT of blocks of $M \times M$ pixels where M is power of 2 are an even function component or odd function component, components are classified into an even/even, an even/odd, an odd/even, and an odd/odd set, and values in a pixel area of $(1/2)M \times (1/2)M$ which correspond to the positions and sizes of coefficients in each set and are put in a table after symmetry is removed are added and cumulated; after the cumulative addition, the cumulative values by the sets are $(1/2)M \times (1/2)M$ of four kinds of sets are added and subtracted by areas obtained by dividing a pixel area of size $M \times M$ for finding a value finally into four.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-306849

(43)公開日 平成7年(1995)11月21日

(51)Int.Cl.
G 0 6 F 17/14

識別記号 庁内整理番号

F I

技術表示箇所

G 0 6 F 15/ 332

S

審査請求 未請求 請求項の数19 O L (全 33 頁)

(21)出願番号 特願平6-167741
 (22)出願日 平成6年(1994)7月20日
 (31)優先権主張番号 特願平6-74121
 (32)優先日 平6(1994)3月18日
 (33)優先権主張国 日本 (JP)

(71)出願人 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中1015番地
 (72)発明者 中川 章
 神奈川県川崎市中原区上小田中1015番地
 富士通株式会社内
 (72)発明者 川勝 保博
 神奈川県川崎市中原区上小田中1015番地
 富士通株式会社内
 (72)発明者 森松 映史
 神奈川県川崎市中原区上小田中1015番地
 富士通株式会社内
 (74)代理人 弁理士 林 恒徳

最終頁に続く

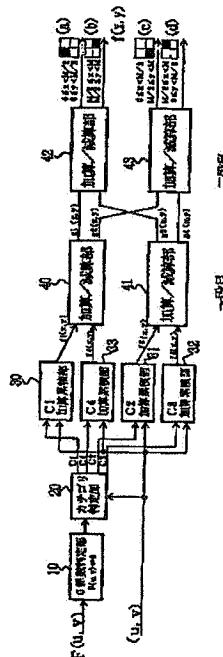
(54)【発明の名称】 二次元逆離散コサイン変換方式

(57)【要約】

【目的】 小さいメモリ容量で復号化処理速度を高速化することができる二次元逆離散コサイン変換方式を提供する。

【構成】 Mが2の巾乗であるM×Mの画素のブロックに対する2次元逆D C Tを行うために対象とする1つのブロックのD C T係数のうち、値が0でない係数が表す基底函数の水平、垂直周波数が、偶函数成分、奇函数成分かに基づき、偶／偶、偶／奇、奇／偶、奇／奇の集合に基づき、各集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた $(1/2) \cdot M \times (1/2) \cdot M$ の画素領域の値を加算累積し、累積加算完了後、各集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を4分割した各領域毎に、4種類の集合の $(1/2) \cdot M \times (1/2) \cdot M$ の大きさの累積値を加算／減算する。

本発明の第一実施例ブロック図



【特許請求の範囲】

【請求項1】Mが2の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、
二次元逆離散コサイン変換を行うべき1つの周波数領域のM×MのブロックのDCT係数のうち、該係数の値が0でないものを検出する手段と、
該検出手段により検出された係数が表す二次元の基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かに基づき、(偶、偶)、(偶、奇)、(奇、偶)、(奇、奇)の4種類の集合に分類する手段と、
該分類手段により分類された各々の集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた $1/2M \times 1/2M$ の画素領域の値を加算累積する手段と、
更に、全ての0でない係数の加算累積完了後、各集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を4分割した各 $1/2M \times 1/2M$ 領域毎に、4種類の集合の $1/2M \times 1/2M$ の大きさの累積値を加算／減算する手段を有して構成されることを特徴とする二次元逆離散コサイン変換方式。

【請求項2】請求項1において、

前記領域を4分割した各 $1/2M \times 1/2M$ 領域毎に、4種類の集合の $1/2M \times 1/2M$ の大きさの累積値の加算・減算を行う手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくするように構成されることを特徴とする二次元逆離散コサイン変換方式。

【請求項3】請求項1において、

前記分類手段により分類される集合は、周波数領域のM×Mの係数の集合を、水平方向、垂直方向とも次数Mの偶の属性を持つ集合と定義し、
ついで、集合の求めかたに関し、それまでに分割された集合の基底が水平周波数、もしくは垂直周波数、もしくはその両方の属性が偶であった場合、その偶であった成分に関し、更に対称性を利用し、基底函数が偶函数である係数の集合と奇函数である係数の集合に分割し、
また、該分割の際に基底函数が偶函数であるものの分割された方向の属性を偶函数、奇函数であるものの集合の属性を奇とし、
該分割された各集合の分割された方向の次数をそれぞれもとの集合の $1/2$ とし、
更に、該分割された各集合に対して、適宜次数が1となるまで該分割を再帰的に行い、該各集合の(水平方向の次数)×(垂直方向の次数)を該集合の画素領域のサイズとするように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項4】請求項3において、前記加算累積手段は、

水平方向、垂直方向とも、前記分割された各集合の(水平方向の次数)×(垂直方向の次数)のサイズの該集合に属する係数の位置と大きさに対応する種類の、画素領域のテーブルを有するように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項5】請求項1乃至4において、前記加算累積手段は、

各集合において、正の係数に対するテーブルのみを有し、負の係数に対しては、該係数の絶対値を用いて、該10テーブルから値を求め、該求められた値を累積値から減算するようにしたことを特徴とする二次元逆離散コサイン変換方式。

【請求項6】請求項1乃至4において、前記加算累積手段は、

各集合もしくは加算段階で求めた中間集合に属する係数が存在するか否かを判断し、その結果に基づいて、それより上位の加算段階で、該集合との加減算を無くするようにして、演算量を削減することを特徴とする二次元逆離散コサイン変換方式。

【請求項7】Mが2の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、

二次元逆離散コサイン変換すべき1つのブロックの二次元離散コサイン変換係数のうち、該変換係数の値が0でないものを判定する手段と、
該変換係数が表す基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かで(偶、偶)、(偶、奇)、(奇、偶)、(奇、奇)の4種類の集合に分類する手段と、

係数が分類された集合ごとに、該集合に属する係数の位置と大きさに対する、対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗算により求める手段と、

該係数が分類された集合ごとに、該 $1/2 M \times 1/2 M$ の画素領域の値を加算して累積する手段と、
全ての0でない係数の累算加算完了後、該4つの集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を4分割した各 $1/2 M \times 1/2 M$ 領域毎に、4種類の集合の $1/2 M \times 1/2 M$ の大きさの累積値の加算・減算を行う手段を有して構成されることを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項8】請求項7において、

前記対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗算により求める手段は、その集合に属する係数ごとに、対応する周波数の係数の大きさ1に対する $1/2 M \times 1/2 M$ 画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずることにより求めることを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項9】請求項7において、

前記対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗

算により求める手段は、 $1/2 M \times 1/2 M$ の各画素ごとに、加算する画素において、該係数が表す縦周波数成分と該係数が表す横周波数成分と、該係数の大きさを乗することにより求めることを特徴とする二次元逆離散コサイン変換方式。

【請求項10】請求項7において、

前記領域を4分割した各 $1/2 M \times 1/2 M$ 領域毎に、4種類の集合の $1/2 M \times 1/2 M$ の大きさの累積値の加算・減算を行う手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくするようにしたことを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項11】Mが2の巾乗である $M \times M$ で構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、

二次元逆離散コサイン変換すべき1つのブロックの二次元離散コサイン変換係数のうち、その値が0でないものを判定する手段と、

該係数が表す基底函数の水平周波数、垂直周波数により、予め決められた種類Nの集合に、係数を分類する手段と、

集合ごとに、係数の位置と大きさに対する、対象性を省いた該係数の属する集合ごとに決められたサイズの画素領域の値を乗算により求める手段と、

集合ごとに、該画素領域の値を加算して累積する手段と、

全ての0でない係数の累積加算完了後、該集合ごとの画素領域の累積値を階層的に加減算を行って画素領域の中間集合を順次求め、かつ、該中間集合も階層的に加減算を行い、最終的に $M \times M$ の大きさの画素領域の値を求める演算を行う手段を有することを特徴とする二次元逆離散コサイン変換方式。

【請求項12】請求項11において、

前記係数を分類すべき、予め決められた種類Nの集合は、第一にもとの $M \times M$ の係数の集合を、水平方向、垂直方向とも次数Mの偶の属性を持つ集合と定義し、次いで、集合の求めかたに関し、それまでに分割された集合の基底が、水平周波数、もしくは垂直周波数、もしくはその両方の属性が偶であった場合、その偶であった成分に関してその更なる対称性を利用し、基底函数が偶函数である係数の集合と奇函数である係数の集合に分割し、また、その際に基底函数が偶函数であるものの集合の分割した方向の属性を偶、奇函数であるものの属性を奇とし、

該分割された各集合の分割された方向の次数をそれぞれもとの集合の1/2とし、更に該分割された各集合に対して、次数が1以上であるかぎり、必要に応じて、該分割を再帰的に行い、該各集合の(水平方向の次数) × (垂直方向の次数)を、該集合の画素領域のサイズとす

るよう構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項13】請求項11において、

前記対象性を省いた、前記係数の属する集合ごとにきめられた画素領域の値を乗算により求める手段は、該集合に属する係数ごとに、該集合の(水平方向の次数) × (垂直方向の次数)のサイズの、係数の大きさ1に対する画素領域のパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずることにより求めるものであることを特徴とする二次元逆離散コサイン変換方式。

【請求項14】請求項11において、

前記対象性を省いた、前記係数の属する集合ごとにきめられた画素領域の値を乗算により求める手段は、該集合に属する係数ごとに縦方向と横方向のパターンを持ち、加算する画素において該係数が表す、該集合の(垂直方向の次数)のサイズの縦周波数成分と該集合の(水平方向の次数)のサイズの横周波数成分と、該係数の大きさを乗ずることにより求めることを特徴とする二次元逆離散コサイン変換方式。

【請求項15】請求項11において、

前記全ての0でない係数の累積加算完了後、該集合毎の累積値を、階層的に加算・減算して $M \times M$ の画素領域の変換値を求める手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくするようにしたことを特徴とする二次元逆離散コサイン変換方式。

【請求項16】Mが2の巾乗である $M \times M$ で構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、

1つのブロック内の非0の係数の数をカウントし、非0の係数の数である有効係数の数を出力する手段と、複数の種類の逆離散コサイン変換を行う手段と、該複数の逆離散コサイン変換を行う手段を選択する手段を有し、

該選択する手段は、該有効係数の数を出力する手段からの有効係数の数により、複数の逆離散コサイン変換を行う手段のいずれかを選択するように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項17】請求項15において、

前記複数の逆離散コサイン変換の少なくとも一つは、有効係数の個数に応じて、逆離散コサイン変換の演算量が変化することを特徴とする二次元逆離散コサイン変換方式。

【請求項18】請求項15において、

前記複数の逆離散コサイン変換の少なくとも一つは、有効係数の個数に関わらず、一定の演算量で逆離散コサイン変換演算を行うものであることを特徴とする二次元逆離散コサイン変換方式。

【請求項19】請求項15において、前記選択する手段は、前記入力された有効係数の個数において、もっとも演算回数が少ない逆離散コサイン変換を行う手段を選択すること特徴とする二次元逆離散コサイン変換方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、画像データの復号化側におけるテーブル参照(ルックアップ)方式を用いる二次元逆離散コサイン変換方式に関する。

【0002】

【従来の技術】大量のデータ、例えば画像データの伝送・蓄積を考える場合、効率において情報量を削減することが不可欠である。そのようなことから、データの相関性を取り除く方式として、様々な方式が検討されてきた。

【0003】その中でも二次元離散コサイン変換(以下、必要な場合、二次元DCTと略称する)およびその逆変換は、それぞれ符号化側、復号化側で画像データに對して用いることにより、画像の相関性を取り除くことができ、直交変換の中でも特に特性が良いことが知られている。

【0004】この様な点から、この二次元離散コサイン変換は、静止画の圧縮の標準化方式であるJPEG、及び動画像符号化の標準化方式であるMPEG、およびH.261に採用されている。

【0005】しかしながら、二次元離散コサイン変換、及びその逆変換は直交変換であり、実現するには行列演算が必要となる。そのため、演算量が非常に大きく、画像のCODEC(符号化/復号化)をハード又はソフトで実現する場合のいずれによっても総演算量の大きな部分を占めている。

【0006】この演算量を削減するため、これまで、ハ

$$f(x, y) = \sum_{u=0}^M \sum_{v=0}^M 1/4C(u)C(v)F(u, v) \cos\left(\frac{(2x+1)u}{2M}\pi\right) \cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

但し、 $C(U) = 1 (u \neq 0), = 1/2^{1/2} (u = 0)$
(定義によって $C(U)$ の値が定数倍だけ異なることもある。)

【0015】上記数1において、 $C(u)$ 、 $C(v)$ は、水平周波数成分及び垂直周波数成分に対するスケールファクタであり、 $F(u, v)$ は、二次元DCTされた係数である。 M は、2の巾乗である。また、 x 、 y は、二次元逆DCTされた画像データの二次元座標である。

【0016】更に、数1について見ると、逆DCT変換

$$f'(x, y, u, v) = 1/4C(u)C(v)F(u, v) \cos\left(\frac{(2x+1)u}{2M}\pi\right) \cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

【0018】従って、 $F(u, v) \neq 0$ の条件を満たす水平周波数及び垂直周波数(u 、 v)についてのみ数2の演算を各 x 、 y に対して行い、その結果を各 x 、 y 毎に加算を行えばよい。

ードウェアで実現する場合には回路規模の小さな方式、ソフトウェアで実現する場合には種々の高速アルゴリズムが検討の対象とされてきた。

【0007】ここで、以下は画像の復号化側で必要となる逆離散コサイン変換に限って議論を進める。従来の高速二次元逆離散コサイン変換(以下、必要な場合、二次元逆DCTと略称する)変換方式の代表例として、Chen等によるアルゴリズムがある(IEEE TRANSACTION ON COMMUNICATION, VOL. COM-25, No. 9, SEPTEMBER, 1977)。

【0008】Chenのアルゴリズムによれば、 8×8 の二次元逆DCTを加算416回、乗算256回で行うことができる。しかしながら、マイクロプロセッサ、DSP等で逆DCT演算を行う場合には十分な演算速度が得られているとは言えない。特に動画データの復号化に用いた場合は、十分なフレームレートが得られず、動画としての使用には耐えない。

【0009】ところで、H.261、MPEG、JPEGを用いた画像符号化標準方式において、伝送されるDCT係数には、以下の3つの性質がある。

【0010】(1) 実用上十分な画質を得られる圧縮率で符号化した場合でも、伝送されるDCT係数のうち、せいぜい $1/8$ が0でない有効な係数であり、残り $7/8$ の係数は0である。

【0011】(2) 0付近の係数の出現確率が非常に高い。

【0012】(3) 係数は量子化されて伝送されるため、DCT係数の全ての位置に対して値が存在するわけではなく、比較的少ない種類の係数のみが出現する。

【0013】ここで二次元逆DCTの定義式を数1のように考える。

【0014】

【数1】

$$f(x, y) = \sum_{u=0}^M \sum_{v=0}^M 1/4C(u)C(v)F(u, v) \cos\left(\frac{(2x+1)u}{2M}\pi\right) \cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

結果 $f(x, y)$ を求める時、 $F(u, v) = 0$ である水平周波数及び垂直周波数(u 、 v)の組み合わせについては、 Σ の内側の数2は0であり、何らの演算も行う必要がないことが分かる。

【0017】

【数2】

$$f'(x, y, u, v) = 1/4C(u)C(v)F(u, v) \cos\left(\frac{(2x+1)u}{2M}\pi\right) \cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

【0019】この結果、 $f(x, y)$ は、次の数3のようになる。

【0020】

【数3】

$$f(x, y) = \sum_{\{u, v\}, F(u, v) \neq 0} f'(F(u, v), x, y, u, v)$$

【0021】ここで、 $f'(level, x, y, u, v)$ [level は係数 $F(u, v)$ のとりうる値] を予めテーブルとして持つておけば、 (x, y) 座標の画素に対して $F(u, v) \neq 0$ の条件を満たす (u, v) についてのみ加算を行えば良い。

【0022】更に、 $M \times M$ のブロックに対して、m個の有効係数（0でない係数）が存在する場合、加算はmが2以上で生ずるので加算の総回数は、 $(m-1) \times M^2$ である。

【0023】このことを、一般的な画像符号化方式に当てはめて検討してみる。MPEGやJPEGでは、一般に1ブロックのサイズは8画素×8画素（M=8）が用いられており、これらの方を用いて一般的な画像を復号した場合、各ブロックの0でない係数、つまり、 $F(u, v) \neq 0$ を満たす係数の数は、平均して8前後であることが知られている。

【0024】この事より、ある座標の画素の値を求めるのに、せいぜい8回の加算のみで値が求まることが分かる。8×8のブロック全体で考えても、448回の加算である。

【0025】一方、従来の高速アルゴリズムの一つであるChenのアルゴリズムでは、8×8の1ブロックあたり加算416回、乗算256回を要する。仮に加算と乗算が同じ速度で処理できるとしても加算換算672回の演算を要する。一般に乗算は、加算に比べて非常に処理に時間を要する。

【0026】更に、 $M \times M$ の二次元逆DCTを行うと、MPEGなどで採用されているレベルがLビットの場合、必要なワード数は、

$(u \text{の種類}) \times (v \text{の種類}) \times (x \text{の種類}) \times (y \text{の種類}) \times (\text{レベルの種類}) = M^4 \times 2^L$ (ワード) となる。

【0027】これは、現在、MPEGなどで一般に用いられている $L=12$ 、 $M=8$ の場合、16Mワードとなる。1ワードを2バイトとすると、必要なテーブルの容量は32Mバイトとなる。このような大きなテーブルを持つには、ハード、ソフトのいずれでも現行技術では実行するのには困難が伴う。

【0028】従って、本出願人は先に、持つべき係数の (u, v) の種類を制限して、逆DCTテーブルサイズの削減を行い、受信した係数に対するテーブルがあればその値を用い、係数に対するテーブルがなければ数2に基づいて値を計算する方式を提案している。

【0029】そして、この方式に基づき逆変換を行う構成の概念ブロック図は、図22に示される。更に、図23はその動作フローチャートである。

【0030】図22において、0係数判別部141によって、入力される基底函数 $F(u, v)$ について、0であるか否かを判定し、 $F(u, v) \neq 0$ である $F(u, v)$ のみを出

力する（ステップS151：図23）。

【0031】ついで、テーブルミス判定部142において、出力される $F(u, v)$ について、 $f'(level, x, y, u, v)$ が予めテーブルとして登録されているか否かを判断し、その結果スイッチSW1及びSW1'を制御する（ステップS152）。

【0032】即ち、テーブルミス判定部142において、0係数判別部141から出力される $F(u, v)$ について、 $f'(level, x, y, u, v)$ が予めテーブルとして登録されている場合は、スイッチSW1及びSW1'を逆DCTテーブル143に接続する。

【0033】したがって、このテーブルから対応する画素領域値 $f'(x, y)$ が outputされる（ステップS153）。また $f'(level, x, y, u, v)$ が予めテーブルとして登録されていない場合は、スイッチSW1及びSW1'を逆DCT演算部144に接続し、画素領域値 $f'(x, y)$ を計算して出力する（ステップS154）。

【0034】このようにして出力される画素領域値 $f'(X, Y)$ が累積加算部145において累積される（ステップS155）。この処理を1ブロック終了するまで継続する（ステップS156）。

【0035】しかしながら、このような本出願人が先に提案する方式においては、テーブルにない係数が来た場合、毎回数2の計算を逆DCT演算部144において行わねばならず、画像の種類によっては、演算量が大幅に増加してしまう虞がある。

【0036】【発明が解決しようとする課題】したがって、本発明の基本的な目的は、小さいメモリ容量で復号化処理速度を高速化することができる画像データ処理における二次元逆離散コサイン変換方式を提供することにある。

【0037】更に本発明の目的は、逆DCTすべき1つのブロックのDCT係数のうち、その値が0でないものをその係数が表す基底函数の水平、垂直周波数がそれぞれ偶函数か奇函数かで集合に分類し、それぞれの集合において、係数の位置と大きさに対応する値をテーブル化して用いるようにする二次元逆離散コサイン変換方式を提供するにある。

【0038】更にまた本発明の目的は、上記周波数成分に対応するテーブルの作成において、その周波数に対し、基本もしくは代表となる係数の値のみテーブルを作成するようにする二次元逆離散コサイン変換方式を提供することにある。

【0039】本発明の更なる目的は、以下の説明及び添付された請求の範囲の記載から明らかになる。

【0040】【課題を解決するための手段】本発明にしたがう二次元逆離散コサイン変換方式の構成は、Mが2の巾乗で

ある $M \times M$ で構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、二次元逆離散コサイン変換を行うべき1つの周波数領域の $M \times M$ のブロックのDCT係数のうち、前記係数の値が0でないものを検出する手段と、前記検出手段により検出された係数が表す二次元の基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かに基づき、(偶、偶)、(偶、奇)、(奇、偶)、(奇、奇)の4種類の集合に分類する手段と、前記分類手段により分類された各々の集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を加算累積する手段と、更に、全ての0でない係数の加算累積完了後、各集合ごとの累積値を、最終的に値を求めるべき $M \times M$ の大きさの画素領域を4分割した各 $1/2 M \times 1/2 M$ 領域毎に、4種類の集合の $1/2 M \times 1/2 M$ の大きさの累積値を加算／減算する手段を有して構成される。

【0041】更に、本発明の上記基本構成と対比される別の構成は、Mが2の巾乗である $M \times M$ で構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、二次元逆離散コサイン変換すべき1つのブロックの二次元離散コサイン変換係数のうち、該変換係数の値が0でないものを判定する手段と、変換係数が表す基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かで(偶、偶)、(偶、奇)、(奇、偶)、(奇、奇)の4種類の集合に分類する手段と、係数が分類された集合ごとに、この集合に属する係数の位置と大きさに対する、対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗算により求める手段と、該係数が分類された集合ごとに、 $1/2 M \times 1/2 M$ の画素領域の値を加算して累積する手段と、全ての0でない係数の累算加算完了後、該4つの集合ごとの累積値を、最終的に値を求めるべき $M \times M$ の大きさの画素領域を4分割した各 $1/2 M \times 1/2 M$ 領域毎に、4種類の集合の $1/2 M \times 1/2 M$ の大きさの累積値の加算・減算を行う手段を有して構成される。

【0042】この別の構成における態様として、上記対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗算により求める手段は、その集合に属する係数ごとに、対応する周波数の係数の大きさ1に対する $1/2 M \times 1/2 M$ 画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずるように構成される。

【0043】

【作用】本発明は、基本的構成において、DCT係数を分類し、その分類された集合ごとに累積加算し、最後に各集合毎の加減算により逆DCTを行う。

【0044】その際、それぞれの集合において、係数の

位置と大きさに対応する対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値が予めテーブル化されている。そして、0でない係数に対応する値をテーブルから読みだし、加算累積する。

【0045】ついで、全ての0でない係数の累積完了後、各集合毎の累積値を、最終的に値を求めるべき $M \times M$ の大きさの画素領域を4分割した各 $1/2 M \times 1/2 M$ 領域毎に、4種類の集合の $1/2 M \times 1/2 M$ の大きさの累積値を加算／減算するよう構成される。

【0046】一方、上記別の構成においては、その集合に属する係数ごとに、集合に属する係数の位置と大きさに対する、対象性を省いた $1/2 M \times 1/2 M$ の画素領域の値を乗算により求め、ついで前記係数が分類された集合ごとに、 $1/2 M \times 1/2 M$ の画素領域の値を加算して累積するようにしている。

【0047】より具体的には、上記別の構成では、その集合に属する係数ごとに、対応する周波数の係数の大きさ1に対する $1/2 M \times 1/2 M$ 画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずるようにしている。したがって、かかる本発明の上記基本構成ならびに別の構成においては、少ないメモリ容量即ち、小さなテーブルサイズで逆DCTのための演算回数の削減と、演算処理の高速化が可能である。

【0048】

【実施例】以下実施例の説明において、同一または、類似のものには同一の数字及び記号を付して説明する。

【0049】図1は、Iピクチャ用のMPEGデコーダであり、本発明の対象とする二次元逆離散コサイン変換方式が適用される部分の位置づけを示すものである。

【0050】まず符号化された画像データのビットストリームは、可変長復号化回路1に入力され、対応するレベルに変換される。次いで量子化回路2において、二次元DCT係数 $F(u, v)$ を出力する。

【0051】二次元DCT係数 $F(u, v)$ は、本発明方式に従う二次元逆DCT回路3により、元の画像の画素データ $f(x, y)$ が再生される。

【0052】図2は、本発明の方式を採用する二次元逆DCT回路3の第一の実施例プロック図である。ここで、図2の実施例の構成及び動作の説明に先立って、本発明の完全な理解のために数的考察を先に行う。

【0053】一般に汎用的な画像符号化その他で用いられる離散コサイン変換の空間の値、及び基底の数は2の巾乗である。このようなことから、Mが2の巾乗である場合を考える。 $0 \leq x, y < M/2$ において定義される以下のような数4を考える。

【0054】

【数4】

11

12

$$f_1(x, y) = \sum_{u=0}^{M/2} \sum_{v=0}^{M/2} \frac{1}{4} C(2u') C(2v') F(2u', 2v') .$$

$$\cos\left(\frac{(2x+1)2u'}{2M}\pi\right) \cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$f_2(x, y)$$

$$= \sum_{u=0}^{M/2} \sum_{v=0}^{M/2} \frac{1}{4} C(2u'+1) C(2v') F(2u'+1, 2v') .$$

$$\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$f_3(x, y)$$

$$= \sum_{u=0}^{M/2} \sum_{v=0}^{M/2} \frac{1}{4} C(2u') C(2v'+1) F(2u', 2v'+1) .$$

$$\cos\left(\frac{(2x+1)2u'}{2M}\pi\right) \cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

$$f_4(x, y)$$

$$= \sum_{u=0}^{M/2} \sum_{v=0}^{M/2} \frac{1}{4} C(2u'+1) C(2v'+1) F(2u'+1, 2v'+1) .$$

$$\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【0055】上記数4から、 f_1 では、水平周波数

($= 2u'$)、垂直周波数 v ($= 2v'$) ともに偶数
(集合C 1)

f_2 では、水平周波数 u ($= 2u' + 1$) が奇数、垂直周波数 v ($= 2v'$) が偶数 (集合C 2)

f_3 では、水平周波数 u ($= 2u'$) が偶数、垂直周波数 v ($= 2v' + 1$) が奇数 (集合C 3)

f_4 では、水平周波数 u ($= 2u' + 1$)、垂直周波数 v ($= 2v' + 1$) ともに奇数 (集合C 4) の成分の函

$$g_1(x, y) = f_1(x, y) + f_4(x, y)$$

$$g_2(x, y) = f_1(x, y) - f_4(x, y)$$

$$g_3(x, y) = f_2(x, y) + f_3(x, y)$$

$$g_4(x, y) = f_2(x, y) - f_3(x, y)$$

二次元離散コサイン変換を行った結果の各画素の値は、

これら式 (1) ~ (4) を用い、またDCT係数の対称性を利用して、以下の様にして求められる。この結果は下記(a) ~ (d) に示した各領域において、数1と等価に

$$f(x, y) = g_1(x, y) + g_3(x, y)$$

$$\dots (1)$$

$$\dots (2)$$

$$\dots (3)$$

$$\dots (4)$$

なる。

【0059】(a) $0 \leq x, y < M/2$ なる (x, y) に対して、

$$\dots (5)$$

に対して、

$$\dots (6)$$

に対して、

$$\dots (7)$$

$$(b) M/2 \leq x < M, 0 \leq y \leq M/2 \text{ なる } (x, y)$$

$$f(x, y) = g_2(M-x, y) - g_4(M-x, y)$$

$$(c) 0 \leq x < M/2, M/2 \leq y < M \text{ なる } (x, y)$$

$$f(x, y) = g_2(x, M-y) + g_4(x, M-y)$$

(d) $M/2 \leq x, y < M$ なる (x, y) に対して、

$$f(x, y) = g_1(M-x, M-y) - g_3(M-x, M-y) \quad \dots \quad (8)$$

以上により、二次元逆離散コサイン変換が表される。但し、上記式(1)～(4)及び上記(5)～(8)に示した $f_1 \sim f_4$ から $f(x, y)$ を求める加算の順序は、あくまでも一例であり、他の加算順序も勿論本発明の範囲から除外されるものではない。

【0060】次に、これらの式を用いた二次元逆離散コサイン変換の方法について、考察する。上記数4の各々の式(1)～(4)を見ると、これは、数1と良く似た形をしている事がわかる。

【0061】このことから、それぞれ式(1)～(4)において、 $F(u, v) \neq 0$ の条件を満たす (u, v) についてのみ数4の各々のΣの中の演算を、各 x, y に対して行い、その結果を各座標の $f(x, y)$ 毎に加算を行えばよい。これを式で表すと次の様になる。

【0062】 f_1 に対しては、 $(u, v) \subseteq C_1$ の係数に対して、数5になる。

【0063】

【数5】

$$f_1(x, y) = \sum_{(u, v), F(u, v) \neq 0} f_1'(F(u, v), x, y, u, v)$$

【0064】 f_2 に対しては、 $(u, v) \subseteq C_2$ の係数に対して、数6になる。

【0065】

【数6】

$$f_2(x, y) = \sum_{(u, v), F(u, v) \neq 0} f_2'(F(u, v), x, y, u, v)$$

【0066】 f_3 に対しては、 $(u, v) \subseteq C_3$ の係数に対して、数7になる。

【0067】

【数7】

$$f_3(x, y) = \sum_{(u, v), F(u, v) \neq 0} f_3'(F(u, v), x, y, u, v)$$

【0068】 f_4 に対しては、 $(u, v) \subseteq C_4$ の係数に対して、数8になる。

【0069】

【数8】

$$f_4(x, y) = \sum_{(u, v), F(u, v) \neq 0} f_4'(F(u, v), x, y, u, v)$$

【0070】ここで、予め $f_1'(\text{level}, x, y, u, v) \sim f_4'(\text{level}, x, y, u, v)$ を計算して、テーブルとして持つておけば、式 $f_1 \sim f_4$ に対しては、それぞれ集合 $C_1 \sim C_4$ に属する係数が $F(u, v) \neq 0$ を満たすものについてのみ、加算を行えば良いことになる。

【0071】式 $f_1 \sim f_4$ が求まれば、上記式(1)～(4)の $g_1 \sim g_4$ 及び式(5)～(8)に従って、画素データである $f(x, y)$ が求まる。

【0072】この手法を用いて、 $M \times M$ の二次元逆DCTを行う場合の必要なテーブルの数、および、演算量に

ついで更に、考察を行う。

【0073】まず、数5～数8を、先に検討した、レベルがLビットの場合を実現する例について考察する。ここでは、 x, y, u, v とともに、 $0 \leq x, y, u, v < M/2$ を満たしている。また、数5～数8に対して、それぞれ独立にテーブルが4種類必要である。これらのことより、数5を実現するために必要なテーブルの大きさは、

$$(\text{テーブルの種類}) \times (\text{uの種類}) \times (\text{vの種類}) \times (\text{xの種類}) \times (\text{yの種類}) \times (\text{レベルの種類}) = 4 \times (M/2)^4 \times 2^L \text{ (ワード)} = M^4 \times 2^{L+3} \text{ (ワード)}$$

【0074】したがって、この場合のテーブルの大きさは、既に検討した本出願人により先に提案している方式の $1/4$ である。

【0075】次に、演算量について考察する。先の仮定と同様に、 $F(u, v) \neq 0$ を満たす係数の個数を m とした場合について考える。ここで、0でない m 個の係数を以下の様に分ける。

【0076】 C_1 に属する係数の個数 : m_1

C_2 に属する係数の個数 : m_2

C_3 に属する係数の個数 : m_3

C_4 に属する係数の個数 : m_4

$$m_1 + m_2 + m_3 + m_4 = m$$

数5～数8において、0でない係数が1つ存在した場合に必要な個数は、

$$(\text{xの種類}) \times (\text{yの種類}) = (M/2)^2$$

30 であるから、数5～数8における必要な加算回数 I_1 は以下のようにになる。

$$\begin{aligned} I_1 &= (M/2)^2 \times (m_1 - 1) + \\ &(M/2)^2 \times (m_2 - 1) + (M/2)^2 \times (m_3 - 1) + (M/2)^2 \times (m_4 - 1) \\ &= (M/2)^2 \times (m_1 + m_2 + m_3 + m_4 - 4) \\ &= 1/4 M^2 \times m - M^2 \end{aligned}$$

【0077】また、前記式(1)～(4)の $g_1 \sim g_4$ を求めるためには、必要な加算の回数 I_2 は、 $I_2 = 4 \times (M/2)^2 = M^2$ となる。

【0078】同様に、上記(a)～(d)を求めるためにも、必要な加算の回数 I_3 は、

$$I_3 = 4 \times (M/2)^2 = M^2$$

【0079】以上のことから、この演算に必要な総加算回数 I は、

$$I = I_1 + I_2 + I_3 = M^2 + 1/4 M^2 \times m$$

【0080】このことから、演算量についても、 $m > 3$ において、既に説明した先に本出願人により提案した方式における演算量である $(m-1) \times M^2$ より削減される。

【0081】上記した本発明の方式を採用する第一の実

施例ブロックが図2乃至図5に示され、その動作フローが図6及び図7に示される。

【0083】以下図2乃至図5を、図6及び図7の動作フローに対応させながら説明する。

【0084】図2において、10は、0係数判定部であり、入力される基底函数の内、 $F(u, v) \neq 0$ である即ち、0でない函数のみ出力する（ステップS101：図6）。

【0085】20は、 $F(u, v) \neq 0$ である函数を分類して出力するカテゴリ判定部であり、その構成例は、図3に示される。即ち、水平及び垂直周波数（u、v）の奇数、偶数を判定して、集合C1～C4として分類し、集合毎に出力する（ステップS201～S203：図6）。

【0086】図3の構成の実現は、マイクロコンピュータ等によりソフト的に実現可能である。ここで集合C1は、先に説明したようにu、vとともに偶数の場合であり、集合C2は、uが奇数で、vが偶数の場合である。

【0087】更に、集合C3は、uが偶数で、vが奇数の場合であり、集合C4は、u、vともに奇数の場合である。

【0088】これら各集合ごとの出力は、C1加算累積部30乃至C4加算累積部33の対応するものに入力され、累積される。C1加算累積部30乃至C4加算累積部33の各々は、同一構成であり、その一例が図4に示される。

【0089】集合Cn（n=1、2、3、4）に属する係数F(u, v)と、水平周波数U、垂直周波数Vが入力される。301は、集合Cnに属するF(u, v)に対応する画素領域の値が登録記憶されるメモリーのテーブルである。

【0090】302は、加算回路であり、303は、累積値を記憶するメモリである。メモリ303の出力が加算回路302に帰還され、テーブル301の出力と再度加算されることにより累積される（ステップS301～S304：図6）。

【0091】そして上記処理が1ブロック分終了するまで繰り返される（ステップS305）。

【0092】図2に戻り、加算累積部30乃至33の出力は、第一段目の加算／減算部40及び41に入力され

る。加算／減算部40には、C1加算累積部30とC4加算累積部33のそれぞれからf1及びf4が入力される。

【0093】一方、加算／減算部41には、C2加算累積部31とC3加算累積部32のそれぞれからf2及びf3が入力される。

【0094】加算／減算部40及び41は、入力されるf1及びf4、f2及びf3に対し、それぞれ加算／減算を行い、g1及びg2、g3及びg4を出力する（ステップS401）。かかる加算／減算は、先に式（1）～（4）のg1乃至g4に関して説明した通りである。

【0095】ここで加算／減算部40、41及び、後に説明する第二段目の加算／減算部42、43は、同一構成であり、その一例が図5に示される。即ち、第一の入力（Inp-1）と第二の入力（Inp-2）は、それぞれ加算回路401及び減算回路402に入力され、第一の入力（Inp-1）と第二の入力（Inp-2）との加算及び減算が行われる。

【0096】加算結果として第一の出力（Out-1）、減算結果として第二の出力（Out-2）が得られる。

【0097】更に、第一段目の加算／減算部40及び41の出力は、第二段目の加算／減算部42、43に入力され、同様に加算／減算される。即ち、加算／減算部42では、第一段目の加算／減算部40からのg1と、加算／減算部41からのg3を加算／減算する。

【0098】一方、加算／減算部43では、第一段目の加算／減算部40からのg2と、加算／減算部41からのg4を加算／減算する。

【0099】この結果、式（1）～（4）のg1～g4に基づき、先に説明した通り、4つの領域（a）乃至（d）に対応する、二次元逆DCTを行った結果f(x, y)が得られる（ステップS402）。

【0100】次に再帰的方法による本発明の第一の実施例の拡張について説明する。

【0101】先ず、u、vとともに偶数成分である、カテゴリC1に属する係数に関する演算である数4のf1(x, y)を書き直すと、数9のとおりになる。

【0102】

【数9】

17

$$f_1(x, y) = \sum_{u'=0}^{M/2} \frac{1}{4} C(2u') \cdot \cos\left(\frac{(2x+1)2u'}{2M}\pi\right).$$

18

$$\sum_{v'=0}^{M/2} F(2u', 2v') \cdot C(2v') \cdot \cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$= \underbrace{\sum_{u'=0}^{M/2} \frac{1}{4} C(2u') \cdot \cos\left(\frac{(2x+1)u'}{2(M/2)}\pi\right)}_{①}.$$

$$\underbrace{\sum_{v'=0}^{M/2} F(2u', 2v') \cdot C(2v') \cdot \cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)}_{②}$$

【0103】ここで数9をよく見ると、部分①、②はともに次数がM/2の逆DCTの式になっていることが分かる。つまり、この数9は数1のMをM/2と変え、
u(=2u')をu'に、v(=2v')をv'に置き換えただけのものと同じ式である。

【0104】したがって、このf1に関しては、
f11(x, y)では、u'、v'ともに偶数（集合C11）
f12(x, y)では、u'が奇数、v'が偶数（集合C12）
f13(x, y)では、u'が偶数、v'が奇数（集合C13）
f14(x, y)では、u'、v'ともに奇数（集合C14）
の様に数1を、数4に分解して式（1）～（4）のg1
～g4によりf(x, y)を求めたと全く同じ方法で、更に

$$\begin{aligned} f_2(x, y) &= \sum_{u'=0}^{M/2} \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot \\ &\quad \sum_{v'=0}^{M/2} F(2u', 2v') \cdot C(2v') \cdot \cos\left(\frac{(2y+1)2v'}{2M}\pi\right) \\ &= \underbrace{\sum_{u'=0}^{M/2} \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)}_{①} \cdot \\ &\quad \underbrace{\sum_{v'=0}^{M/2} F(2u', 2v') \cdot C(2v') \cdot \cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)}_{②} \end{aligned}$$

【0108】この数10をみると、偶数成分であるv'に関しては、部分②の様に次数M/2の逆DCTの式となるが、奇数成分であるu'に関しては、部分①の様にカテゴリC1の場合と異なり、次数M/2の逆DCTの式とはならない。この場合には、数10は、以下のように分

f1(x, y)に属する係数をf11～f14のように4つの集合に分割し、f1(x, y)を求めるための演算量、テーブルサ

イズを小さくすることが可能である。

【0105】また、f11(x, y)、f12(x, y)、f13(x, y)は分割した基底の両方もしくは片方が偶数成分であるため、さらなる分割も可能である。

【0106】次に、uが奇数成分、vが偶数成分である、カテゴリC2に属する係数に関する演算である数4のf2(x, y)を書き直すと、数10のとおりになる。

【0107】

【数10】

解できる。

【0109】奇数のv'に対しては、数11のとおりになる。

【0110】

【数11】

19

20

$$h21(x, y) = \sum_{u'=-M/2}^{M/2} \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot$$

$$\sum_{v'=-M/4}^{M/4} F(2u', 4v') \cdot C(4v') \cdot \cos\left(\frac{(2y+1)2v'}{2(M/2)}\pi\right)$$

【0111】偶数の v' に対しては、数 12 のとおりに
なる。

【0112】

【数 12】

$$h22(x, y) = \sum_{u'=-M/2}^{M/2} \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot$$

$$\sum_{v'=-M/4}^{M/4} F(2u', 2(2v'+1)) \cdot C(2(2v'+1)) \cdot \cos\left(\frac{(2y+1)(2v'+1)}{2(M/2)}\pi\right)$$

【0113】そして、(e) $0 \leq x < M/2$ 、 $0 \leq y < M/4$ なる (x, y) に対して、
 $f2(x, y) = h21(x, y) + h22(x, y) \dots (9)$

(f) $0 \leq x < M/2$ 、 $M/4 \leq y < M/2$ なる (x, y) に対して、
 $f2(x, y) = h21(x, M/4-y) - h22(x, M/4-y) \dots (10)$

なる式が表現できる。

【0114】このことから、この場合も、数 5～数 8 と
同様にテーブルを用いて $h21(x, y)$, $h22(x, y)$ を求め、
これから上記式 (9)、(10) を用いて $f2(x, y)$ を求
めることができる。

【0115】この場合も、テーブルサイズ、演算量と
も、数 6 を用いて求めるより削減されている。この場
合、まず $h21(x, y)$, $h22(x, y)$ から $f2(x, y)$ を求め、あ
とは、先に示した式 (1)～(4) 及び式 (5)～(1
0) に示した方法で $f(x, y)$ が求まる。

【0116】また、 $h21(x, y)$ に関しては、 v'' 成分が
偶数成分なので、さらなる分割が可能である。

【0117】以上に示した様に、その基底が偶函数の集
合に対して、それらの基底は次数を半分にした DCT の
基底と等価となるので、その集合のさらなる分割が可
能となる。

【0118】この様に、基底が偶函数の成分に関して
は、更にその係数の集合を分割でき、その結果、演算量
とテーブルサイズともにより小さくできる。また、この

分割は基底が偶数であるかぎり必要に応じて再帰的に行
うことが可能である。

【0119】次に本発明の第二の実施例について説明す
る。

【0120】一般に、係数である Level $F(u, v)$ は、正
と負のどちらも存在する。また、Level が正負反転する
と、 $f1'(level, x, y, u, v) \sim f4'(level, x, y, u, v)$ も、
正負反転する。このことから、 $f1'(level, x, y, u, v) \sim$
 $f4'(level, x, y, u, v)$ に関しても、level が正の値、も
しくは負の値に対する、どちらか一方のみ用意すればよ
い。

30 【0121】この場合において、 $f1 \sim f4$ を求める方
法を以下に示す。

【0122】 $n=1, 2, 3, 4$ として、 f_n は、
 $(u, v) \subseteq C_n$ の係数に対して、 $f_n(x, y)$ は、数 1
3 のとおりである。

【0123】

【数 13】

$$fn(x, y) = \sum_{(u, v), F(u, v) > 0} fn'(|F(u, v)|, x, y, u, v)$$

$$- \sum_{(u, v), F(u, v) < 0} fn'(|F(u, v)|, x, y, u, v)$$

【0124】この数 13 を用いると、
 $1/2 \times (\text{テーブルの種類}) \times (u \text{の種類}) \times (v \text{の種類}) \times$
 $(x \text{の種類}) \times (y \text{の種類}) \times (\text{レベルの種類}) = 1/2 \times$
 $4 \times (M/2)^4 \times 2^4 \text{ (ワード)} = M^4 \times 2^{1-3} \text{ (ワ
ード)}$
 となる。

【0125】これは先に考察した演算量 $(M^4 \times 2$

$^{1-2})$ に対し、更に半減する。したがって、既に説明し
た、本出願人が先に提案した図 22、図 23 に示す方式
に比べ $1/8$ となり、大幅にテーブルサイズが削減で
きる。この方法により、更に必要とするテーブルの大
きさを半分にする事が可能となる。

【0126】さらに、DCT 係数が C1～C4 のうち、
 C1, C2, C4 に属するもののみが存在し、C3 に属

するものが存在しない場合を考える。この場合、 $f_3(x, y)$ は0である。この場合、 $g_3(x, y)$ 、 $g_4(x, y)$ は、単に $f_2(x, y)$ をコピーすれば良いだけで、加算が必要なくなる。

【0127】C3に加えて、C2に属する係数も存在しない場合は、もはや、 $g_3(x, y)$ 、 $g_4(x, y)$ も0で、その結果、 g_3 、 g_4 を求める為の加算は必要でなく、さらに、 $f(x, y)$ も単に、 $g_1(x, y)$ 、 $g_2(x, y)$ のコピーだけで値が求まる。

【0128】このように、予め、加算を行う際に、片方の加算すべき集合に属する係数が存在するかどうかを基に条件判断し、適応的に加算を行うことにより、さらに平均的な加算回数を削減することが可能となる。

【0129】この、適応的に加減算を行う方法、および、正負のテーブルをまとめてテーブルサイズを小さくする方法を用いた、本発明方式の第二実施例ブロック図を図8～図10に、その動作フローチャートを図11～図14に示す。

【0130】第一の実施例の場合と同様に実施例ブロック図を動作フローチャートと対象しながら説明する。

【0131】図8において、0係数判定部10及びカテゴリ判定部20の構成及び動作は、図2に示す場合と同様であり、0係数の判定（ステップS101：図11）及びカテゴリの判定が行われる（ステップS201乃至S203）。

【0132】カテゴリ判定部20の出力は、対応する累積加算部30乃至33に入力される。そこで累積加算される。第二の実施例における累積加算部30乃至33の構成例が図9に示される。図13は、累積加算部30乃至33の対応する累積加算動作フローである。

【0133】図9において、 $F(u, v)$ として有効係数が一度でも入力される場合は、スイッチ307により“1”にセットする（ステップS132）。即ちスイッチ307により、“1”が出力されるように制御する。

【0134】絶対値化部304によりカテゴリ判定部20の出力 $F(u, v)$ が、絶対値化される。ついで、入力される $F(u, v)$ が正であるか否かが判定される（ステップS132）。

【0135】テーブル305は、 $f'_n(level, u, v, x, y)$ 、但し $level > 0$ に対する画素領域の値を登録記憶しており、絶対値化された $F(u, v)$ により対応する画素領域の値が読みだされる。

【0136】306は、加算／減算回路であり、テーブル305の出力に対し、 $F(u, v)$ が正であれば、そのまま累積加算値を記憶するメモリ303の出力と加算し、負で有れば、極性を反転してメモリ303の出力と加算する（ステップS133、S134）。

【0137】これを x, y のループの終わりまで継続する（ステップS135、S136）。したがって、テーブル305は、 $F(u, v)$ により対応する画素領域の値と

して、正負ある画素領域の値の内、正の画素領域の値のみ記憶すればよくメモリ容量が削減出来る。

【0138】そして、上記の処理は1ブロック終了するまで継続される（ステップS305）。

【0139】各累積加算部30乃至33から有効係数の有無により出力される $e_1 \sim e_4$ は、一段目の適応的加算／減算部50、51及びオアゲート回路60、61に入力される。

【0140】即ち、累積加算部30及び33からの出力 e_1 及び e_4 は、適応的加算／減算部50及びオアゲート回路60に入力し、累積加算部31及び32からの出力 e_2 及び e_3 は、適応的加算／減算部51及びオアゲート回路61に入力する。

【0141】一段目の適応的加算／減算部50、51及び、それらの出力及びオアゲート60、61の出力が入力される二段目の適応的加算／減算部52、53の構成例は、図10に示される。即ち、第一及び第二の入力 I_{np-1}, I_{np-2} 及び入力 S_1 と S_2 の組み合わせに対応して、加算又は減算を行うように構成される演算回路である。また、図14にその動作フローが示される。

【0142】図14において、 e_n, e_m をそれぞれ S_1, S_2 とすると、入力 S_1 と S_2 とともに“1”である場合（ステップS141:y, S142:y）、第一の出力として第一及び第二の入力 I_{np-1}, I_{np-2} の和が、第二の出力として第一及び第二の入力 I_{np-1}, I_{np-2} の差が演算され、出力される（ステップS144）。

【0143】更に、入力 S_1 が“1”、入力 S_2 が“0”である場合（ステップS141:y, S142:n）、第一及び第二の出力ともに第一の入力 I_{np-1} が出力される（ステップS145）。

【0144】また、入力 S_1 が“0”であり、入力 S_2 が“1”である場合（ステップS141:n, ステップS143:y）、第一の出力には I_{np-2} 、第二の出力には I_{np-2} に(-1)乗じたものが出力される（ステップS146）。更に、入力 S_1 と S_2 とともに“0”である場合（ステップS141:n, S143:n）、第一及び第二の出力ともに0が出力される（ステップS147）。

【0145】そして、それぞれの状態において、 x, y のループが終わるまで上記処理が続き、それぞれ $f_1(x, y) \sim f_4(x, y)$ を出力する（ステップS148～S151）。

【0146】即ち、上記のように加算累積のステップS301～304の後、1ブロック終了するまで次の画素の処理を継続する（ステップS305）。

【0147】図11及び図12に戻り、図8の実施例においての具体的な適応的加算／減算部50乃至53の動作について説明する。

23

【0148】それぞれの加算累積部30乃至33からの $f_1(x, y) \sim f_4(x, y)$ 出力は、対応する第一段目の適応的加算／減算部50、51及びオアゲート60、61に入力される。

【0149】適応的加算／減算部50について考察すると、S1、S2としてe1とe4が入力され、同時に第一、第二の入力として、 $f_1(x, y)$ 及び $f_4(x, y)$ が入力される。したがって、これらe1とe4により $f_1(x, y)$ 及び $f_4(x, y)$ の適応的加算／減算が行われ、 $g_1(x, y)$ 、 $g_2(x, y)$ を出力する（ステップS501：図12）。

【0150】適応的加算／減算部51についても同様に処理が行われ、 $g_3(x, y)$ 、 $g_4(x, y)$ を出力する（ステップS502）。

【0151】e1とe4は、オアゲート60に入力され、その論理和e5がオアゲート60から出力される。同様に、e2とe3は、オアゲート61に入力され、その論理和e6がオアゲート61から出力される（ステップS601）。

【0152】次いで、e5とe6をS1、S2として第二段目の適応的加算／減算部52、53に入力する。また、 $g_1(x, y)$ と $g_3(x, y)$ が適応的加算／減算部52に、 $g_2(x, y)$ と $g_4(x, y)$ が適応的加算／減算部53に入力する。

【0153】そして、適応的加算／減算部52において、e5とe6による $g_1(x, y)$ と $g_3(x, y)$ の加算／減算が行われ、適応的加算／減算部53において、e5とe6による $g_2(x, y)$ と $g_4(x, y)$ の加算／減算が行われ、第fn' (level1, x, y, u, v) =

$$\frac{1}{4}C(u)C(v)(level1*2+1)\cos\left(\frac{(2x+1)2}{16}\pi\right)\cos\left(\frac{(2y+1)2}{16}\pi\right)$$

ただし、 $0 \leq x, y < 8$ 、 $-1024 \leq level1 < 1023$ である

【0161】このテーブルサイズは、
 $1/2 \times (\text{テーブルの種類}) \times (uの種類) \times (vの種類) \times (xの種類) \times (yの種類) \times (\text{レベルの種類}) = 1/2 \times 4 \times (8/2)^4 \times 2^{12}$ (ワード) $\times 2$ バイト = 4194304
 バイト
 である。

【0162】これは、現在のパーソナルコンピュータで十分実現可能な大きさである。MPEGにおいては、逆量子化して得られる係数は奇数のみであるので、エンタリイとしては1以上の奇数のみを用意すればよい。本実施例では、

`level1 = int(F(u, v) / 2)`
 (但し、int(x)は小数点以下を切り捨てる。) として、level1を用いて、各カテゴリにおいてテー

24

一の実施例の場合と同様にそれぞれ $f(x, y)$ を出力する（ステップS503、S504）。

【0154】尚、ステップS503、S504の加減算処理は、先に説明した式(5)～(9)にしたがって、アドレスを入れ換えて行われる。

【0155】図15は、本発明の方式と先に説明した既に提案した方式との演算回数の比較を示す図である。係数の個数を横軸として、縦軸に既に提案したテーブルを用いた二次元逆離散コサイン変換と本発明方式の加算回数を示す。

【0156】本発明の方式においては、条件判断により加算を適応的に行い、さらに、それぞれの係数の個数に対する最悪値（なるべく多くのC1～C4の集合に係数が割り振られた場合）を示している。この様に、全ての係数の個数において、本方式の方が、総演算回数が少ないことが分かる。

【0157】ここで、説明の簡単化のため1ビクチャのみ復号できるものとし、M=8に相当するMPEGを考える。上記実施例で用いたブロック図及び動作フローを用い、M=8と置く。

【0158】更に、二次元逆離散コサイン変換は16ビット固定小数点で行うものとする。

【0159】かかる場合、4種類のテーブル $f_1' \sim f_4'$ は、それぞれ以下のように構成される。 $(u, v) \in C$ nの係数に対して、数14のよう表される。

【0160】

【数14】

ブルルックアップを行うことにより、逆DCTの計算がされる。

【0163】この場合の演算量は、係数が8個存在する場合、全ての演算量は192回であり、従来の方式に比べて、加算の演算量が大幅に削減されている。

【0164】更に、上記第一の実施例の方式を用いて、上記C1を4分割、C2、C3を2分割した場合、必要なテーブルは、2359296バイトとなり、更にテーブルサイズが小さくなり、その分演算量も削減される。

【0165】これまで説明した本発明の実施例では、係数の周波数成分と大きさの両方に關してテーブルを作成するものとして説明した。ここで、周波数成分が変わると、コサイン基底の乗算が必要で、浮動点少數演算が必要となるが、係数の大きさに關していえば、整数しか存

在しえない。

【0166】このようなことから、本発明の第三の実施例として、各集合のテーブルを作成する際に、周波数成分のみに関してテーブルを持ち、レベル $F(u, v)$ に関しては、その値を乗することにより画素領域の値を求め、その後は、既に説明した手順に基づき逆DCTを行うことも可能である。

【0167】かかる本発明の第三の実施例について、以下詳細に説明する。

【0168】先ず第一の実施例と同様に、第三の実施例

$$f1'(x, y, u, v) = \frac{1}{4}C(2u')C(2v') \cdot \cos\left(\frac{(2x+1)2u'}{2M}\pi\right) \cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

【0171】集合 C_2 では、 $u (= 2u' + 1)$ 、 $v (= 2v')$ 、 x, y に対して数 16 のように定義される。

$$f2'(x, y, u, v) = \frac{1}{4}C(2u'+1)C(2v') \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

【0173】集合 C_3 では、 $u (= 2u')$ 、 $v (= 2v' + 1)$ 、 x, y に対して数 17 のように定義される。

$$f3'(x, y, u, v) = \frac{1}{4}C(2u')C(2v'+1) \cdot \cos\left(\frac{(2x+1)2u'}{2M}\pi\right) \cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【0175】また集合 C_4 では、 $u (= 2u' + 1)$ 、 $v (= 2v' + 1)$ 、 x, y に対して数 18 のように定義される。

$$f4'(x, y, u, v) = \frac{1}{4}C(2u'+1)C(2v'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【0177】そして、 $F(u, v) \neq 0$ の条件を満たす (u, v) についてのみ、以下の数 19、数 20、数 21、数 22 のように、集合 $C_1 \sim C_4$ の中からこの係数の属するものを選択し、その領域に対して、係数の値を、先に数 15～数 18 に示した、 $f1'(x, y, u, v) \sim f4'(x, y, u, v)$ のテーブルの値に乘じて

$$f1(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f1'(x, y, u, v)$$

【0180】 f_2 に対しては、 $(u, v) \subseteq C_2$ の係数に対して、数 20 となる。

$$f2(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f2'(x, y, u, v)$$

【0182】 f_3 に対しては、 $(u, v) \subseteq C_3$ の係数に対して、数 21 となる。

$$f3(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f3'(x, y, u, v)$$

【0184】 f_4 に対しては、 $(u, v) \subseteq C_4$ の係数 50 に対して、数 22 となる。

における二次元逆離散コサイン変換の手法について考察する。

【0169】上記数 4 から求められる集合 $C_1 \sim C_4$ に属する各係数と $0 \leq x, y < M/2$ なる x, y に対して定義されるテーブル $f1' \sim f4'$ を以下の様に定義する。即ち、集合 C_1 では、 $u (= 2u')$ 、 $v (= 2v')$ 、 x, y に対して数 15 のように定義される。

【0170】

【数 15】

【0172】

【数 16】

【0174】

【数 17】

【0176】

【数 18】

加算を行って、 $f1 \sim f4$ を求める。

【0178】 f_1 に対しては、 $(u, v) \subseteq C_1$ の係数に対して、数 19 となる。

【0179】

【数 19】

【0181】

【数 20】

【0183】

【数 21】

【0185】

$$f4(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f4'(x, y, u, v)$$

【0186】以上の処理を、全ての $F(u, v) \neq 0$ を満たす係数に対して行えば、 $f1 \sim f4$ が求まり、さらに $f(x, y)$ が求まって、逆DCTが完了する。

【0187】ここで、予め $f1'(x, y, u, v) \sim f4'(x, y, u, v)$ を計算して、テーブルとして持つておくことにより、 $f1 \sim f4$ に対しては、それぞれ集合 $C1 \sim C4$ に属する係数が、 $F(u, v) \neq 0$ を満たすものについてのみ、係数の値を $f1'(x, y, u, v) \sim f4'(x, y, u, v)$ のテーブルの値に乗じて加算を行えば良いことになる。

【0188】 $f1 \sim f4$ が求まれば、式(1)～式(8)に従って画素データである $f(x, y)$ が求まる。

【0189】この第三の実施例の手法を用いて $M \times M$ の二次元逆DCTを行う場合に必要なテーブルの数、および、演算量について考察を行う。

【0190】まず、数15～数18を、先に検討したレベルがレビットの場合の実現について考察する。ここでは、 x, y, u, v とともに、 $0 \leq x, y, u, v < M/2$ を満たしている。

【0191】また、数15～数18に対して、それぞれ独立にテーブルが4種類必要である。これらのことより、数15～数18を実現するために必要なテーブルの大きさは、(テーブルの種類) \times (uの種類) \times (vの種類) \times (xの種類) \times (yの種類) $= 4 \times (M/2)^4 = M^4 / 4$ (ワード)

となる。 $M=8$ で考えると高々1024ワードである。

【0192】次に、演算量について考察する。先の仮定と同様に、 $F(u, v) \neq 0$ を満たす係数の個数をmとした場合について考える。ここで、0でないm個の係数を以下の様に分ける。

【0193】集合 $C1$ に属する係数の個数 : m_1

集合 $C2$ に属する係数の個数 : m_2

集合 $C3$ に属する係数の個数 : m_3

集合 $C4$ に属する係数の個数 : m_4

ここで、 $m_1 + m_2 + m_3 + m_4 = m$ である。

【0194】したがって、加算の演算量については、先に説明したテーブルルックアップ方式と同様である。これにより、演算量も従来より削減される。

【0195】また、同様な計算で乗算量Jは、

$$J = 1/4 M^2 \times (m_1 + m_2 + m_3 + m_4)$$

$$= 1/4 M^2 \times m$$

である。

【0196】上記の第三の実施例の実現する構成ブロック内の、基本的概念ブロック構成は、第二の実施例の構成ブロック図である図8と同様に示されるので、再度の

【数22】

図示は、省略する。

【0197】更に、他の実施例と異なる特徴部分のブロック図は、図16に示され、更にその動作の特徴とする部分のフローチャートが図17、図18に示される。

【0198】他の実施例の場合と同様に実施例ブロック図を動作フローチャートと対照しながら説明する。

10 【0199】図8を参照すると、0係数判定部10及びカテゴリ判定部20の構成及び動作は、図2に示す場合と同様であり、0係数の判定(例えば、図11のステップS101参照)及びカテゴリの判定が行われる(例えば、図11のステップS201乃至S203参照)。

【0200】カテゴリ判定部20の出力は、対応する累積加算部30乃至33に入力され、累積加算される。第三の実施例における累積加算部30乃至33の構成例が図16に示される。

【0201】図17は、累積加算部30乃至33の対応する累積加算動作フローである。

【0202】図16において、 $F(u, v)$ として有効係数が一度でも入力される場合は、スイッチ307により“1”にセットする(ステップS181)。即ちスイッチ307により、“1”が出力されるように制御する。

【0203】テーブル308は、先に説明した実施例と異なり、レベルに関する情報を含まない周波数成分 $f'n(u, v, x, y)$ を登録記憶しており、集合 Cn に属する (u, v) 及び (x, y) により対応する画素領域の値が読みださられる。

【0204】この読みだされた $f'n(u, v, x, y)$ と、 $F(u, v)$ が乗算器309により乗算され、対象性を省いた $1/2M \times 1/2M$ の画素領域の値が求められる。

【0205】ついで、乗算器309の出力は、累積加算器303からの出力と加算器302において加算される(ステップS182)。これを x, y のループの終わりまで継続する(ステップS183)。

【0206】これにより、それぞれの係数が分類された集合ごとに、上記 $1/2M \times 1/2M$ の画素領域の値を加算して累積される。

【0207】したがって、テーブル308は、 $F(u, v)$ により対応する画素領域の値として、周波数成分 $f'n(u, v, x, y)$ のみを記憶すればよくメモリ容量が削減出来る。

【0208】各累積加算部30乃至33から有効係数の有無により出力される $e1 \sim e4$ は、一段目の適応的加算／減算部50、51及びオアゲート回路60、61に入力される。

【0209】即ち、累積加算部30及び33からの出力 $e1$ 及び $e4$ は、適応的加算／減算部50及びオアゲー

ト回路60に入力し、累積加算部31及び32からの出力e2及びe3は、適応的加算／減算部51及びオアゲート回路61に入力する。

【0210】一段目の適応的加算／減算部50、51及び、それらの出力及びオアゲート60、61の出力が入力される二段目の適応的加算／減算部52、53の構成例は、図10に示されるものと同様である。

【0211】即ち、第一及び第二の入力Inp-1、Inp-2及び入力S1(en)とS2(em)の組み合せに対応して、加算又は減算を行うように構成される演算回路である。また、図18にその動作フローが示される。

【0212】図18において、en、emをそれぞれS1、S2とすると、入力S1とS2がともに“1”である場合(ステップS191:y, S192:y)、第一の出力として第一及び第二の入力Inp-1、Inp-2の和が、第二の出力として第一及び第二の入力Inp-1、Inp-2の差が演算され、出力される(ステップS194)。

【0213】更に、入力S1(en)のみが“1”である場合(ステップS191:y, S192:n)、第一及び第二の出力ともに第一の入力Inp-1が出力され

$$f1(x, y) =$$

$$\sum_{u=0}^{M/2} \sum_{v=0}^{M/2} \frac{1}{4} F(2u', 2v') C(2u') C(2v') \cos\left(\frac{(2x+1)u'}{2(M/2)} \pi\right) \cos\left(\frac{(2y+1)v'}{2(M/2)} \pi\right)$$

【0219】この数23の式を良く見ると、次数がM/2の逆DCTの式になっていることが分かる。つまり、この式は、定係数以外は、数1のMをM/2に変え、u' (= 2u') をu'に、v' (= 2v') をv'に置き換えただけのものと同じ式である。

【0220】つまり、このf1に関しては、
f11(x, y)では、u', v'ともに偶数(集合C1)
f12(x, y)では、u'が奇数、v'が偶数(集合C12)
f13(x, y)では、u'が偶数、v'が奇数(集合C13)
f14(x, y)では、u', v'ともに奇数(集合C14)

の様に数1を数4の各項に分解して、式(1)～式

る(ステップS195)。

【0214】また、入力S1(en)が“0”であり、入力S2(em)が“1”である場合(ステップS191:n, S192:y)、第一の出力にはInp-2、第二の出力にはInp-2に(-1)乗じたものが出力される(ステップS196)。

【0215】更に、入力S1(en)とS2(em)がともに“0”である場合(ステップS191:n, S192:n)、第一及び第二の出力とともに0が出力される(ステップS197)。

【0216】そして、それぞれの状態において、x, yのループが終わるまで上記処理が続き、それぞれf1(x, y)～f4(x, y)を出力する(ステップS198～S201)。

【0217】次に、第三の実施例の拡張として、再帰的な手法について考察する。先ず、ここで、u, vともに偶数成分である、カテゴリC1に属する係数に関する演算である上記数4の第一式、f1(X, Y)を書き直すと、数23のごとくなる。

【0218】

【数23】

(8) の方法で f1(x, y) を求めたのと全く同じ方法により、さらに f1(x, y) に属する係数を f11～f14 のように 4 つの集合へ分割する。

【0221】これにより、f1(x, y) を求めるための演算量、テーブルサイズを小さくすることが可能である。

【0222】また、f11(x, y)、f12(x, y)、f13(x, y)は、分割した基底の両方もしくは片方が偶成分であるため、さらなる分割も可能である。

【0223】次に、uが奇数成分、vが偶数成分である、集合C2に属する係数に関する演算である数4の第40二項を書き直すと、数24の如くになる。

【0224】

【数24】

$$\begin{aligned}
 f2(x, y) &= \sum_{u'=0}^{M/2} \sum_{v'=0}^{M/2} F(2u', 2v') \cdot \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \\
 &\quad \cdot C(2v') \cdot \cos\left(\frac{(2y+1)2v'}{2M}\pi\right) \\
 &= \sum_{u'=0}^{M/2} \sum_{v'=0}^{M/2} F(2u', 2v') \cdot \frac{1}{4} \underbrace{C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)}_{①} \\
 &\quad \cdot C(2v') \cdot \cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right) \\
 &\quad \underbrace{\qquad\qquad}_{②}
 \end{aligned}$$

【0225】この数24の式をみると、偶数成分である
vに関しては②の様に次数M/2の1次元逆DCTの式
となるが、奇数成分であるuに関しては集合C1の場合
と異なり、次数M/2の逆DCTの式とはならない。

【0226】この場合には、この式は以下の様に分解で
きる。ここでは、以下の様に、集合C2をC21、C2
の2個の集合に分割する。

$$\begin{aligned}
 h21(x, y) &= \sum_{u'=0}^{M/2} \sum_{v''=0}^{M/4} F(2u', 4v'') \cdot \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \\
 &\quad \cdot C(4v'') \cdot \cos\left(\frac{(2y+1)2v''}{2(M/2)}\pi\right)
 \end{aligned}$$

【0229】また、奇数のv' ($\cong C22$) に対して
は、下記数26となる。

$$\begin{aligned}
 h22(x, y) &= \sum_{u'=0}^{M/2} \sum_{v''=0}^{M/4} F(2u', 4v'') \cdot \frac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot \\
 &\quad C(2(2v''+1)) \cdot \cos\left(\frac{(2y+1)(2v''+1)}{2(M/2)}\pi\right)
 \end{aligned}$$

【0231】そして、

(e) $0 \leq x < M/2$, $0 \leq y < M/4$ なる (x, y)

に対して、

$$f2(x, y) = h21(x, y) + h22(x, y) \quad \dots \quad (9)$$

(f) $0 \leq x < M/2$, $M/4 \leq y < M/2$ なる (x, y) に対して、

$$f2(x, y) = h21(x, M/4-y) - h22(x, M/4-y) \quad \dots \quad (10)$$

なる式で表現できる。

【0232】したがって、この場合も、上記数15～数
18と同様に以下の様なテーブルを定義する。即ち、偶
数のv' (= 2v'') に対しては、下記数27を定義す

る。

【0233】

【数27】

$h21'(x, y, u, v) =$

$$\frac{1}{4}C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot C(4v'') \cdot \cos\left(\frac{(2y+1)2v''}{2(M/2)}\pi\right)$$

【0234】奇数の v' ($= 2v'' + 1$) に対しては、下記数 28 を定義する。

 $h22'(x, y, u, v) =$

$$\frac{1}{4}C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)$$

$$\cdot C(2(2v''+1)) \cdot \cos\left(\frac{(2y+1)(2v''+1)}{2(M/2)}\pi\right)$$

【0236】そして、以下の式の様に、係数が属する領域に対して、係数の値を、先に示した数 27、数 28 の $h21'(x, y, u, v)$ 及び $h22'(x, y, u, v)$ のテーブルの値に乗じて加算を行い、 $h21 \sim h22$ を求める。

$$h21(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot h21'(x, y, u, v)$$

【0239】 $h22$ に対しては、 $(u, v) \subseteq C22$ の係数に対して、下記数 30 となる。

$$h22(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot h22'(x, y, u, v)$$

【0241】この様に、テーブルと乗算を用いて $h21(x, y)$ 、 $h22(x, y)$ を求める。次いで上記式 (9)、(10) を用いて $f2(x, y)$ を求めることができる。この場合も、テーブルサイズ、演算量とも、上記数 16 を用いて求めるより削減されている。

【0242】この場合、まず $h21(x, y)$ 、 $h22(x, y)$ から $f2(x, y)$ を求める。その後、式 1 ~ 式 8 に示した方法で $f(x, y)$ が求まる。

【0243】また、 $h21(x, y)$ に関しては、 v'' 成分が偶成分なので、さらなる分割が可能である。

【0244】以上に示した様に、その基底が偶函数の集合に対して、それらの基底は、次数を半分にした DCT の基底と等価となるので、その集合の更なる分割が可能となる。この様に、基底が偶函数の成分に関しては、更にその係数の集合を分割でき、その結果、演算量とテーブルサイズともにより小さくできる。

【0245】また、この分割は、必要な限り、もしくは基底が偶数であるかぎり再帰的に行うことが可能である。

【0246】上記第三の実施例の拡張の考え方において、更に、 x, y, u, v でテーブルを持つ代わりに、水平成分の (x, u) 、垂直成分の (y, v) の組ごとにテーブルを独立に持つ（例えば、集合 $C21, C22$ においては、上記数 24 の①と②）、それらのテーブルの値を乗じて加算すべき値を求めるようにすることが可

【0235】
【数 28】

【0237】 $h21$ に対しては、 $(u, v) \subseteq C21$ の係数に対して、下記数 29 になり、

【0238】
【数 29】

能である。

【0247】この場合は、乗算回数は倍になるが、必要とするテーブルサイズが小さくなる。更に、上記した予め、加算を行う際に片方の加算すべき集合に属する係數が存在するかどうかを元に条件判断し、適応的に加算を行うことにより、更に加算回数を削減する考えかたと同様に、まず、N 個の各領域に有効係数があったかどうかのフラグを基に、それらの領域してひとつ広い領域の値を求める際に適応的に加算する。

【0248】そして、その新しい領域のフラグは、その領域を求めるのに用いた 2 つの領域の少なくとも何方か一方に有効係数があった場合に有効とする。この処理を、領域が $M \times M$ になるまで行うようにすることが可能である。

【0249】次に、第四の実施例として上記第三の実施例において、テーブル 308 に $f1'(x, y, u, v) \sim f4'(x, y, u, v)$ のテーブルデータを持つ代わりに、以下の様なテーブルを定義することが可能である。即ち、 $0 \leq x < M/2$ 、奇数の u に対し、数 31 を定義する。

【0250】
【数 31】

$$fe(x, u) = \frac{1}{2}C(u)\cos\left(\frac{(2x+1)u}{2M}\pi\right)$$

【0251】又、 $0 \leq x < M/2$ 、偶数のuに対し、数32を定義する。

【0252】

【数32】

$$f_0(x, u) = \frac{1}{2} C(u) \cos\left(\frac{(2x+1)u}{2M}\pi\right)$$

$$f_1(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0256】 f_2 に対しては、 $(u, v) \subseteq C_2$ の係数 10 【0257】
に対して、数34を定義する。 【数34】

$$f_2(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0258】 f_3 に対しては、 $(u, v) \subseteq C_3$ の係数 10 【0259】
に対して、数35を定義する。 【数35】

$$f_3(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0260】 f_4 に対しては、 $(u, v) \subseteq C_4$ の係数 10 【0261】
に対して、数36を定義する。 【数36】

$$f_4(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0262】この後は、式(1)～式(8)と同じ方法で $f(x, y)$ を同様に求まる。即ち、 $1/2 M \times 1/2 M$ の画素ごとに、加算する画素において、その係数が表す縦周波数成分(u)と横周波数成分()と、その係数の大きさを乗算することにより求める方法である。

【0263】この第四の実施例方法での演算量は、第三の実施例に比べて加算の回数は変わらないものの、乗算の回数に関しては、2倍の $J=M^2/2$ となる。

【0264】しかしながら、この手法のテーブルサイズは、

$$\begin{aligned} & (x \text{の種類}) \times (u \text{の種類}) \times (\text{テーブルの種類}) \\ & = (M/2) \times (M/2) \times 2 \\ & = M^2/2 \end{aligned}$$

となり、第三の実施例に比べて、テーブルサイズが大幅に削減される。この第四の実施例における加算累積部30～33の構成を図19に示す。

【0265】図19は、図16との比較において、テーブル308が第一及び第二のテーブル308a、308bに分割されている。第一のテーブル308aは、垂直周波数成分(u)をテーブルとして記憶するものである。

【0266】第二のテーブル308bは、水平周波数成分()をテーブルとして記憶するものである。したがって、更に乗算器310を設け、乗算器309と協働して、上記数33～数36の各々の乗算を行う。

【0267】更に、これら乗算の結果は、加算器302と、累積加算器303により累積上記数33～数36の各々の累積加算を行う。

【0253】したがって、 $f_1 \sim f_4$ は、以下の様にして求まる。

【0254】 f_1 に対しては、 $(u, v) \subseteq C_1$ の係数に対して、数33を定義する。

【0255】

【数33】

$$f_1(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0256】 f_2 に対しては、 $(u, v) \subseteq C_2$ の係数 10 【0257】
に対して、数34を定義する。 【数34】

$$f_2(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0258】 f_3 に対しては、 $(u, v) \subseteq C_3$ の係数 10 【0259】
に対して、数35を定義する。 【数35】

$$f_3(x, y) = \sum_{(u, v), F(u, v) \neq 0} F(u, v) \cdot f_0(x, u) \cdot f_0(y, v)$$

【0260】 f_4 に対しては、 $(u, v) \subseteq C_4$ の係数 10 【0261】
に対して、数36を定義する。 【数36】

【0268】更に、DCT係数が集合C1～C4のうち、C1, C2, C4に属するもののみが存在し、C3に属するものが存在しない場合を考える。この場合、 $f_3(x, y)$ は0である。この場合、 $g_3(x, y)$ 、 $g_4(x, y)$ は、単に $f_2(x, y)$ をコピーすれば良いだけで、加算が必要なくなる(式(3)、(4)参照)。

30 【0269】C3に加えて、C2に属する係数も存在しない場合は、もはや、 $g_3(x, y)$ 、 $g_4(x, y)$ も0で、その結果 g_3 、 g_4 を求める為の加算は必要なく、さらに、 $f(x, y)$ も単に、 $g_1(x, y)$ 、 $g_2(x, y)$ のコピーだけで値が求まる。

【0270】このように、予め、加算を行う際に、片方の加算すべき集合に属する係数が存在するかどうかを基に条件判断し、適応的に加算を行うことにより、さらに平均的な加算回数を削減することが可能となる。

【0271】この片方の加算すべき集合に属する係数が存在するかどうかの判断は、0係数判定部10で行われる。

【0272】次に、上記の第三の実施例以降の実施例において具体的な数字を当てはめて本発明の効果を考察する。MPEGやJPEG、H.261などの標準化で用いられている $M=8$ の場合を考える。

【0273】第三の実施例を考え、まずテーブルサイズを求める。1ワードあたり2バイトと仮定し、先の示した $M'/4$ (ワード)のテーブルサイズの大きさに $M=8$ を代入すると、2048バイト(2Kバイト)となる。インテル社製の80486DX2のMPUは8Kバ

イトのキャッシュを内蔵しているが、この大きさのテーブルなら全てがキャッシュ上に格納でき、メインメモリ上に大きなテーブルを専有することなく高速演算が期待できる。

【0274】また、1ブロックに8個の有効係数があるブロックの逆変換を考える。集合に属する係数毎に、対応する周波数の基本的な（つまり、係数の大きさ1に対する） $1/2 M \times 1/2 M$ 画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを求める場合には、乗算回数は、128回、加算回数が256回となる。これは、明らかにChenの方式などより少ない。

【0275】また、第三の実施例の拡張手法を用いると、乗算回数が倍になり256回となる。この回数はChenの方式と同じであるが、CPUでこの方式を実現した場合、アルゴリズムが単純な分だけ、Chenの方式よりも高速化が期待できる。またこの時のテーブルサイズは、先の場合と同様に1ワードが2バイトと仮定すれば、32バイトとなっており、必要なテーブルサイズがさらに小さくなる。

【0276】また、第四の実施例を用いて、C1をさらに4分割、C2、C3をそれぞれさらに2分割した場合、テーブルサイズは、1ワード2バイトとして、1152バイトとなる。

【0277】この場合、必要なテーブルサイズが先の2048バイトに比べてもおよそ半分になっている。またそれに応じてアルゴリズムは若干複雑になるものの、必要な演算回数もより少なくすることが可能である。

【0278】次に本発明の第五の実施例として、複数の二次元逆離散コサイン変換のうち最適なものを選んで、最悪値見積もりが可能な二次元逆離散コサイン変換を実現する方式を説明する。

【0279】上記実施例に従い説明したように、本発明の改良型テーブルルックアップ方式により有効係数が少ない時には、従来から知られているどの方式よりも高速に二次元逆離散コサイン変換を行うことが可能である。

【0280】既に検討したように、特に係数の数が17より少ないときは、Chenの手法よりも乗算回数を削減することができ、また係数の数が9より少ない時は、現在知られているなかで最も乗算回数が少ないHagueの手法により、乗算回数を削減することができる。

【0281】しかしながら、本発明の手法では、係数の数が多いときには、従来の高速演算より逆に演算量が増加してしまう。また、演算速度が係数の数に左右されてしまうため、演算速度の最悪値の見積もりが困難である。

【0282】したがって、本発明の第五の実施例は、かかる問題を解決するものである。

【0283】図20は、第五の実施例の基本概念を説明するブロック図である。従来の高速手法は、係数の数に

よらず演算速度が一定である。また、上記本発明の実施例では、係数が少ないと高速であるものの、係数の個数が多いときには演算量が増大してしまう。

【0284】そのため、予めブロックの有効係数の数を数え、その個数の時に、最も高速な手法を選ぶことにより、係数が多いブロックに対しても、最悪値を極端に増大させることなく逆離散コサイン変換を可能とする。

【0285】図20において、210は、有効係数の数を数えるカウンタであり、220は、カウンタ210からの有効係数の数を入力し、離散コサイン変換を行う方式を選択する離散コサイン変換選択部である。

【0286】上記のように各方式を採用する場合において、有利な有効係数の数が求められるので、離散コサイン変換選択部220に有利な有効係数の数と対応づけて記憶して置く。

【0287】231～23nは、複数の離散コサイン変換方式の各々に対応し、演算処理を行う演算回路である。離散コサイン変換選択部220は、カウンタ210から入力される有効係数の数を参照して、有利な離散コサイン変換を行う方式を選択して切替え信号を入力スイッチSW1、出力スイッチSW2に送る。

【0288】したがって、入力スイッチSW1、出力スイッチSW2により切替え接続された演算回路231～23nのいずれかに有効係数が入力される。

【0289】このように、予めブロックの有効係数の数をカウントし、その個数に対応する最も有利な方式を選択することにより、係数の多いブロックに対しても、最悪値を極端に増大させることなく逆離散コサイン変換を実行することが可能である。

【0290】図21は、図20に対応する第五の実施例ブロック図であり、 8×8 の逆離散コサイン変換を行う回路である。図21においては、本発明とChenの方法を切り換える実施例である。

【0291】M=8として、 8×8 のブロックがカウンタ210に入力する。カウンタ210は、有効係数をカウントし、カウントした有効係数の個数を離散コサイン変換選択部220に入力する。

【0292】離散コサイン変換選択部220には、有効係数の個数nとして、 $n > 16$ の時、Chen方式を対応づけ、 $n \leq 16$ の時、本発明の演算手法に対応付けるべくテーブルが記憶されている。

【0293】したがって、演算回路として本発明の改良型テーブルルックアップ方式による演算回路231と、Chenのアルゴリズムによる演算回路232を有する。

【0294】スイッチSW1、SW2により有効係数カウンタ210において、有効係数の個数nとして、 $n > 16$ の時、演算回路232に、 $n \leq 16$ の時、演算回路231に切り換えられる。これにより、全体として、高速な二次元逆離散コサイン変換が実現される。

【0295】

【発明の効果】本発明によれば、二次元離散コサイン変換とその逆変換を用いたシステムにおいて、小さなテープルサイズで復号処理速度を高速化することができる。特にマイクロプロセッサを用いて構成された、画像復号器に適用した場合に効果が大きい。

【0296】また、上記の通り実施例にしたがい本発明を説明したが、かかる実施例は、本発明の説明のためのものであり、本発明は、かかる実施例に限定されるものではない。

【0297】本発明の技術思想と一致するもの及び均等の範囲にあるものは、本発明の保護の範囲に含まれるものである。

【図面の簡単な説明】

【図1】Iピクチャ用MPEGデコーダの構成図である。

【図2】本発明の第一の実施例ブロック図である。

【図3】図2におけるカテゴリ判定部の構成例である。

【図4】図2における加算累積部の構成例である。

【図5】図2における加算／減算部の構成例である。

【図6】第一の実施例動作フロー（その1）である。

【図7】第一の実施例動作フロー（その2）である。

【図8】本発明の第二の実施例ブロック図である。

【図9】図8における加算累積部の構成例である。

【図10】図8における加算／減算部の構成例である。

【図11】第二の実施例動作フロー（その1）である。

【図12】第二の実施例動作フロー（その2）である。

【図13】第二の実施例動作フロー（その3）である。

【図14】第二の実施例動作フロー（その4）である。

【図15】本発明の効果を説明する図である。

【図16】本発明の第三の実施例における加算累積部の構成例である。

【図17】図16における加算累積部の動作フローである。

【図18】本発明の第三の実施例の拡張における適応的加算／減算のフローチャートである。

【図19】本発明の第四の実施例における加算累積部の構成例である。

【図20】本発明の第五の実施例の原理ブロック図である。

【図21】本発明の第五の実施例の具体的実施例である。

【図22】先に提案された逆DCT方式のブロック図である。

【図23】先に提案された逆DCT方式の動作フローである。

【符号の説明】

1 可変復号化回路

2 逆量子化回路

3 二次元逆離散コサイン変換回路

10 0係数判定部

20 20 カテゴリ判定部

31～32 加算累積部

40～43 加算／減算部

50～53 適応的加算／減算部

60、61 オアゲート

301、305、308、308a、308b IDC

T係数テーブル

302、401 加算回路

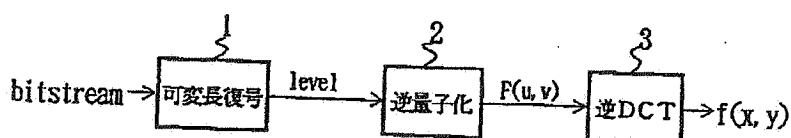
303 累積加算メモリ

402 減算回路

30 304 絶対値化回路

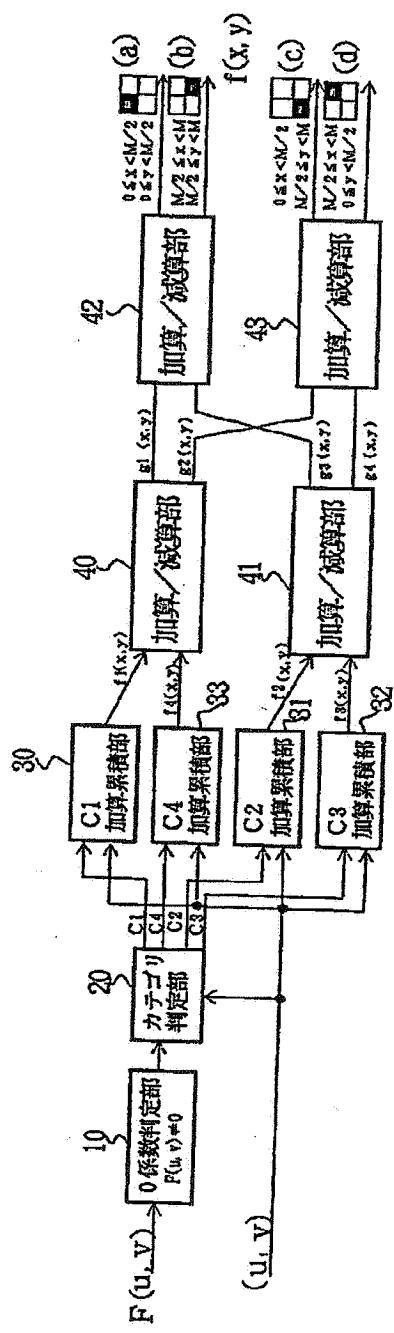
306 加算／減算回路

309、310 乗算器

【図1】**Iピクチャ用MPEGデコーダの構成図**

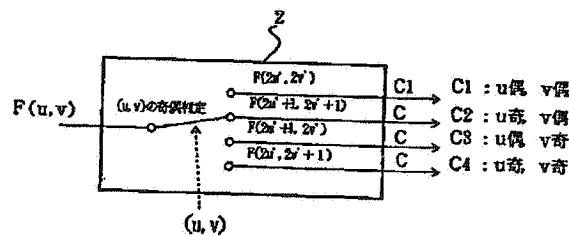
【図2】

本発明の第一実施例ブロック図

一段目
二段目

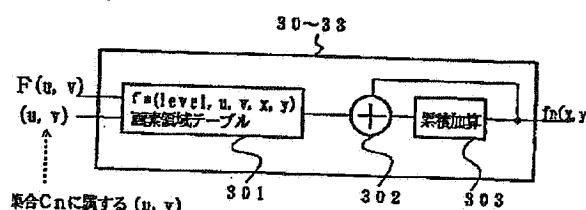
【図3】

カテゴリ判定部の構成例



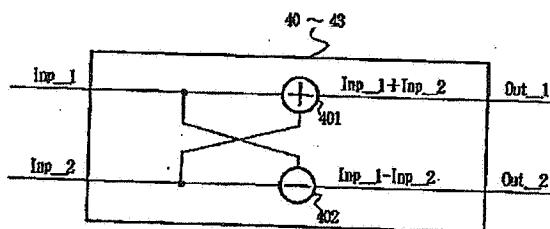
【図4】

加算累積部の構成例



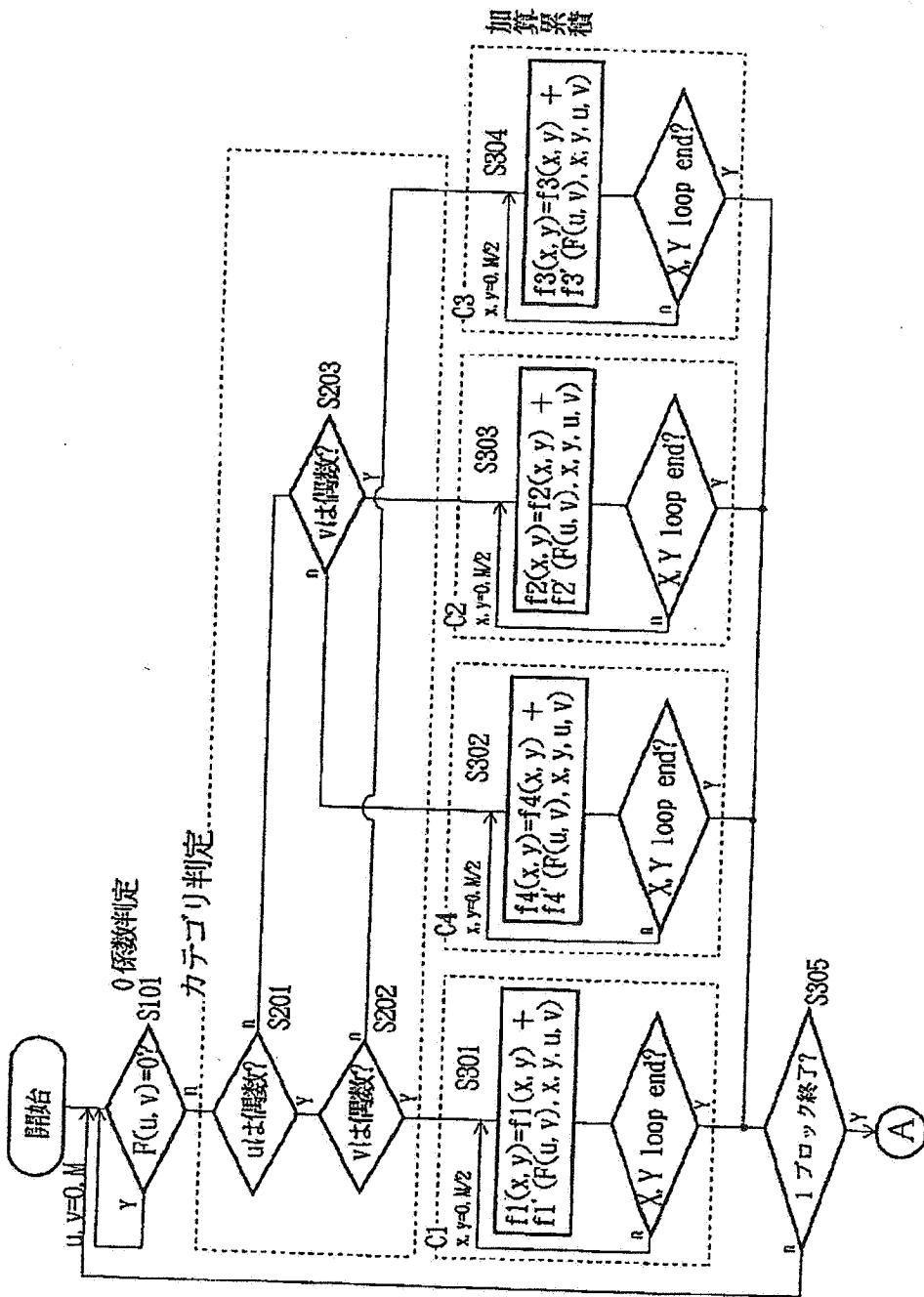
【図5】

加算/減算部の構成例



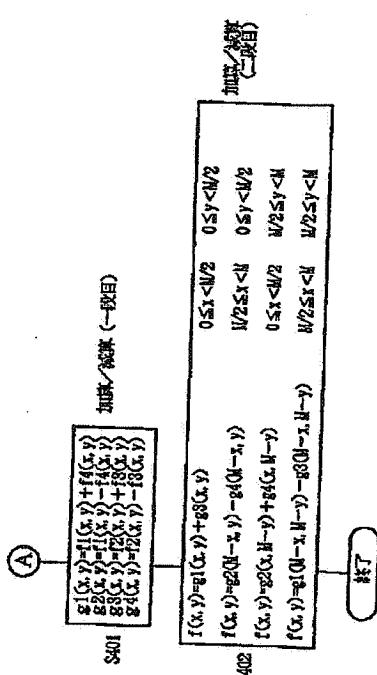
【図6】

第一の実施例動作フロー（その1）



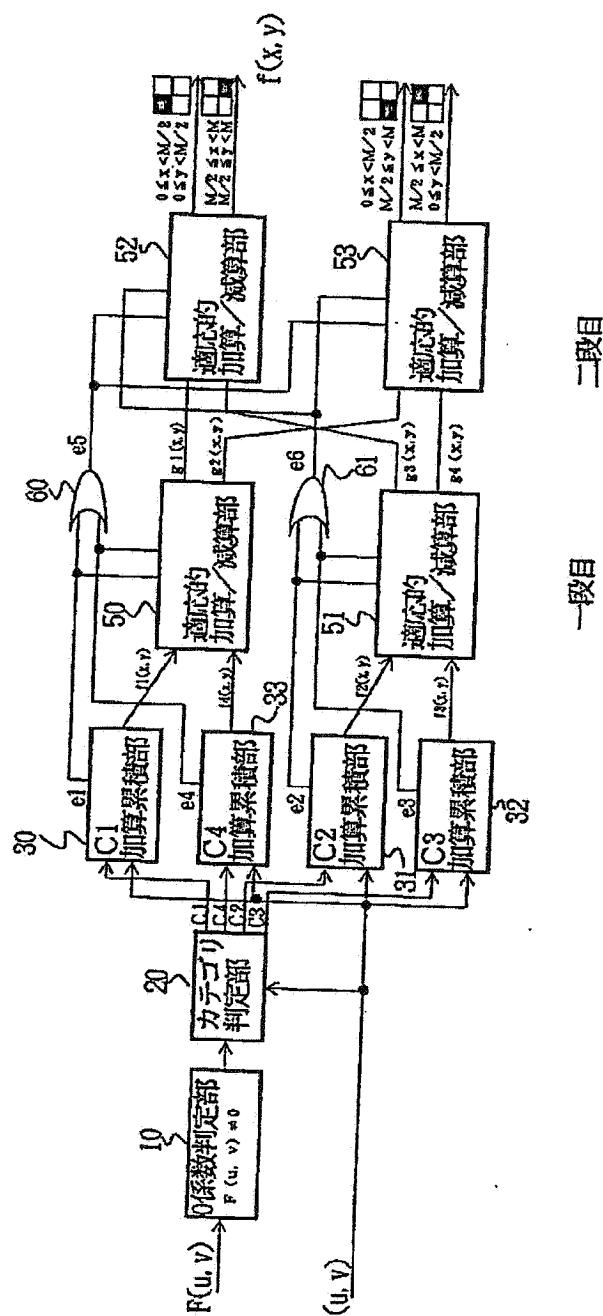
【図7】

第一の実施例動作フロー(その2)

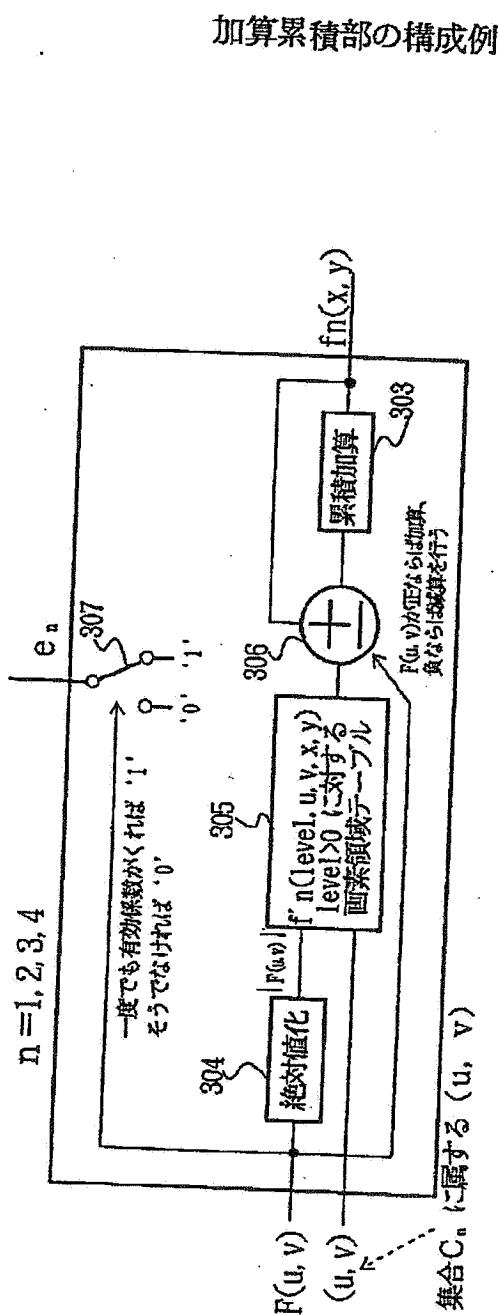


【図8】

本発明の第二の実施例ブロック図

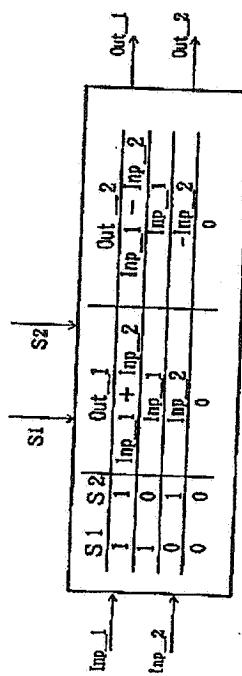


【図9】



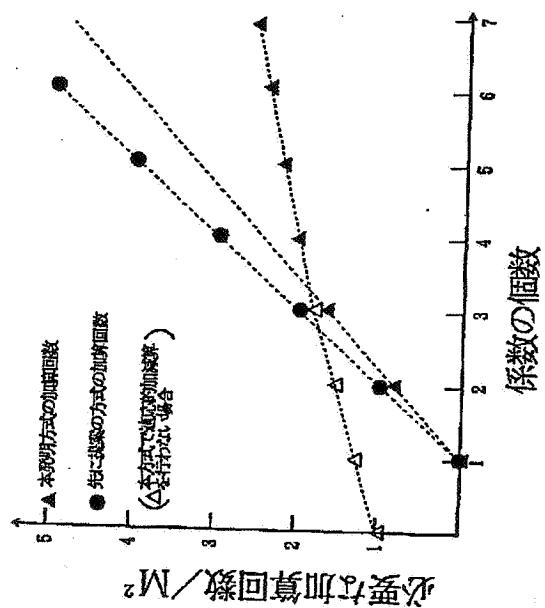
【図10】

加算／減算部の構成例



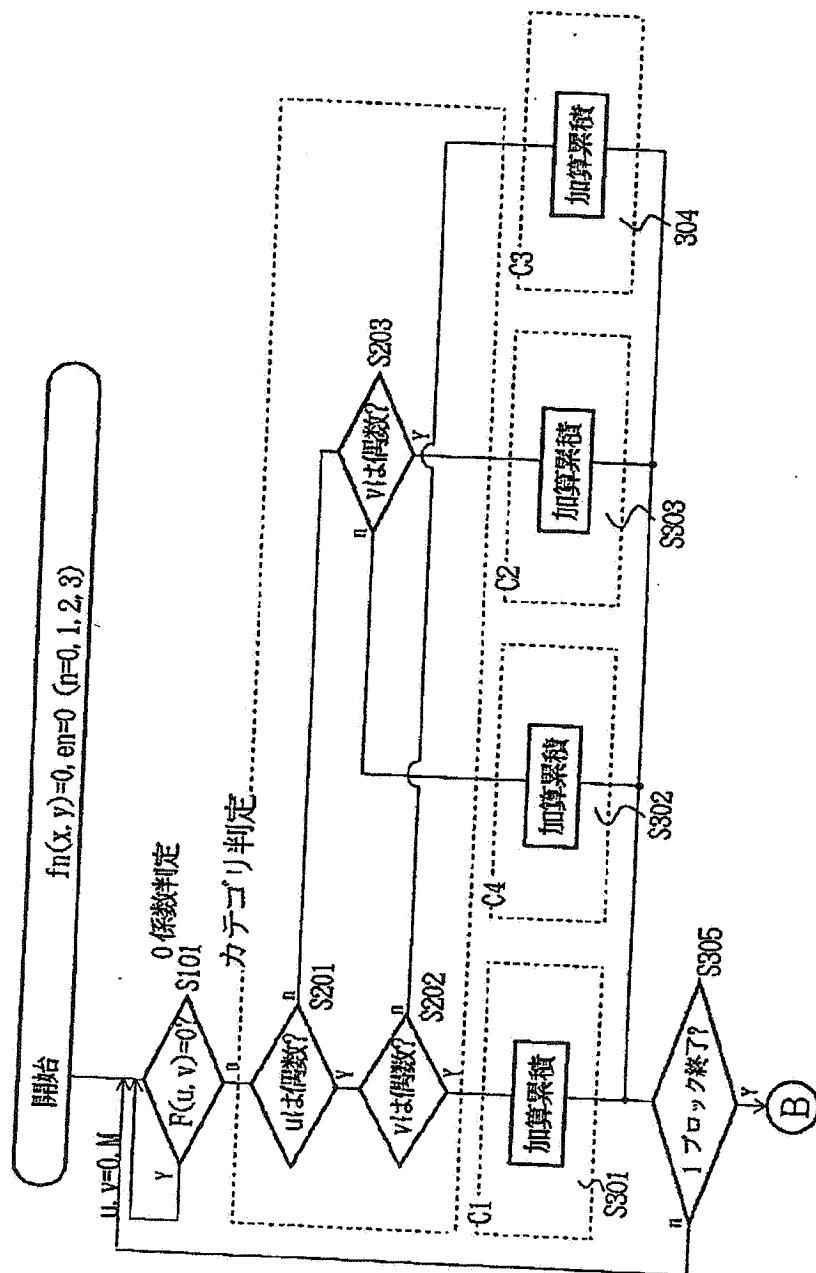
【図15】

本発明の効果を説明する図



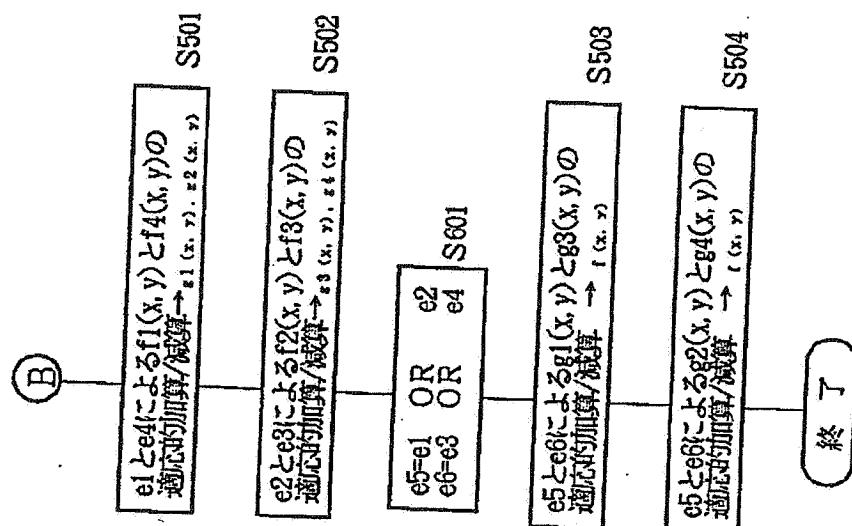
【図11】

第二の実施例動作フロー（その1）



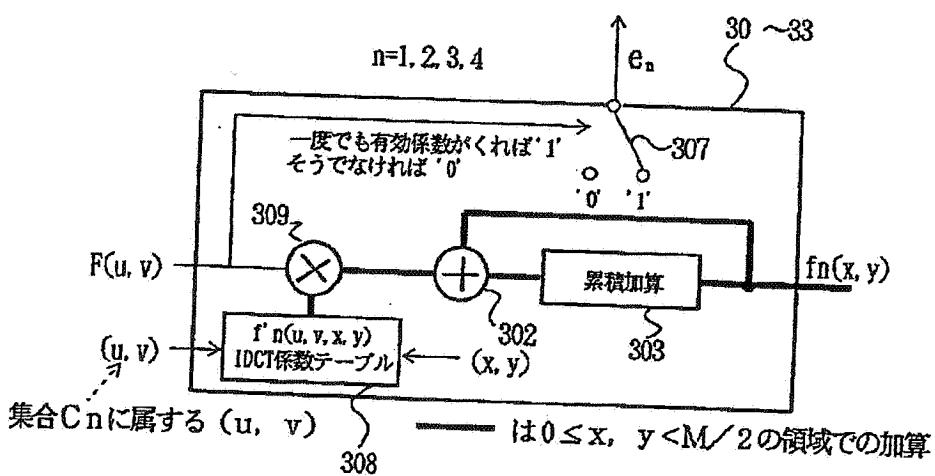
【図 12】

第二の実施例動作フロー（その 2）



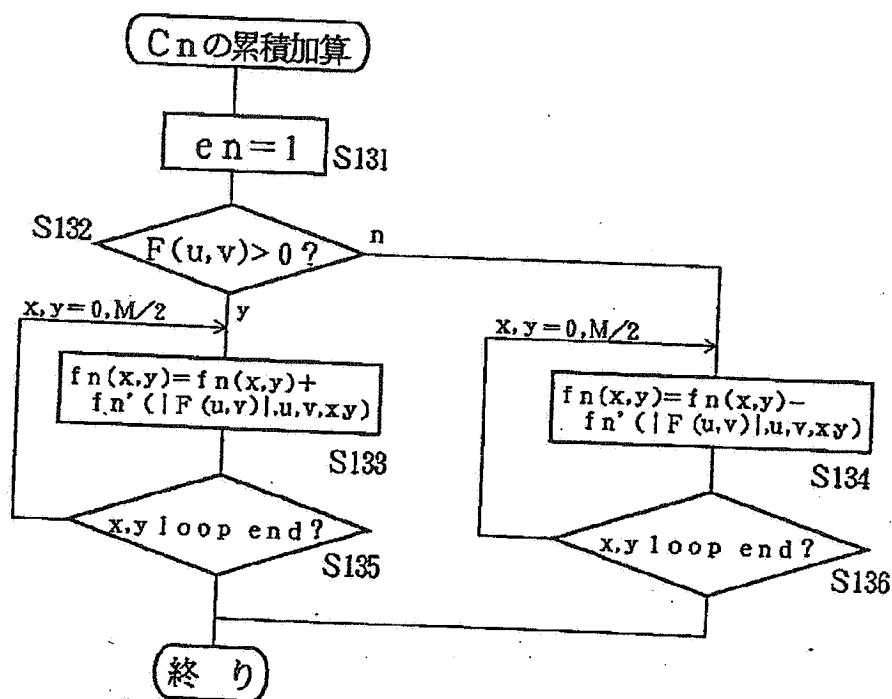
【図 16】

加算累積部の構成例



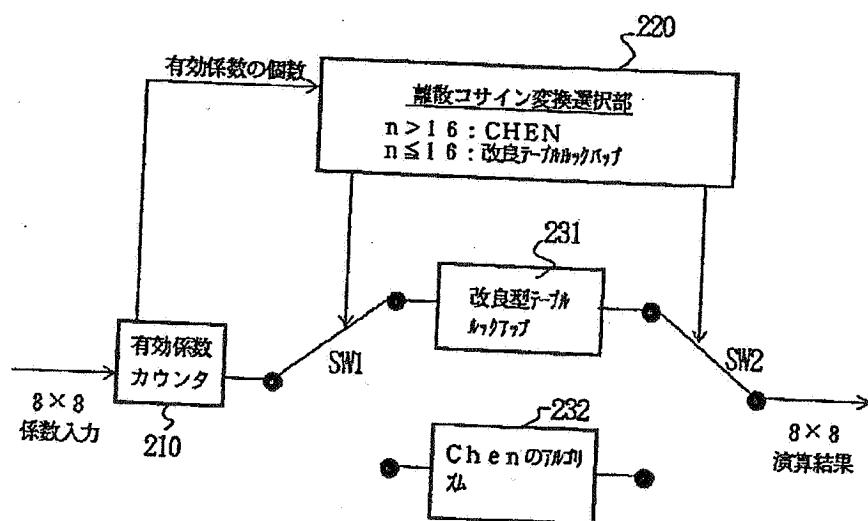
【図13】

第二の実施例動作フロー (その3)



【図21】

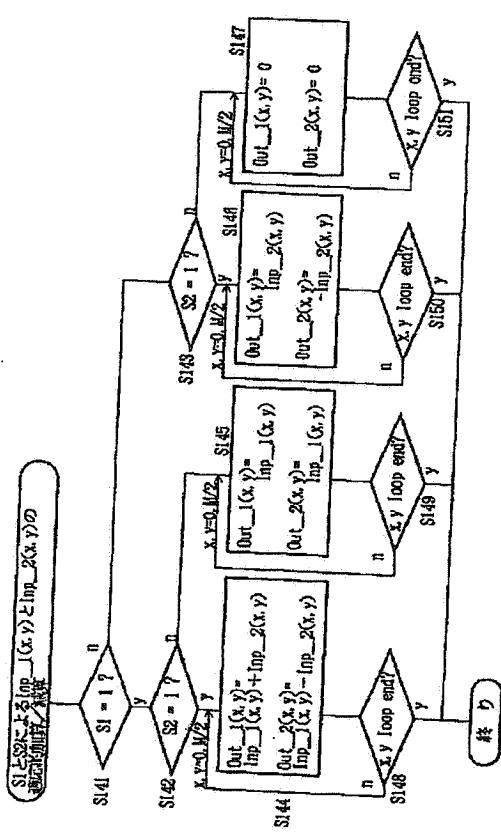
本発明の第五の実施例の具体的な実施例



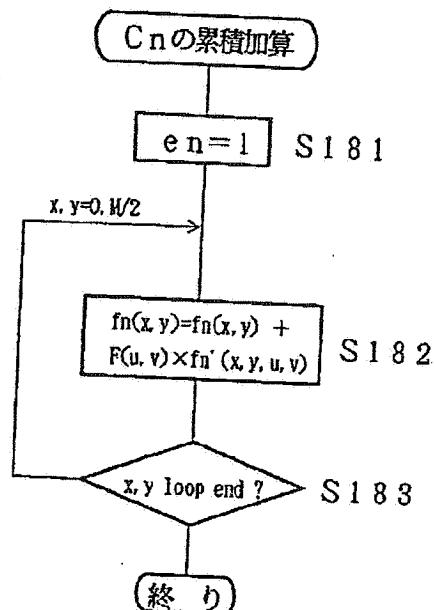
【図14】

[図17]

第二の実施例（その4）

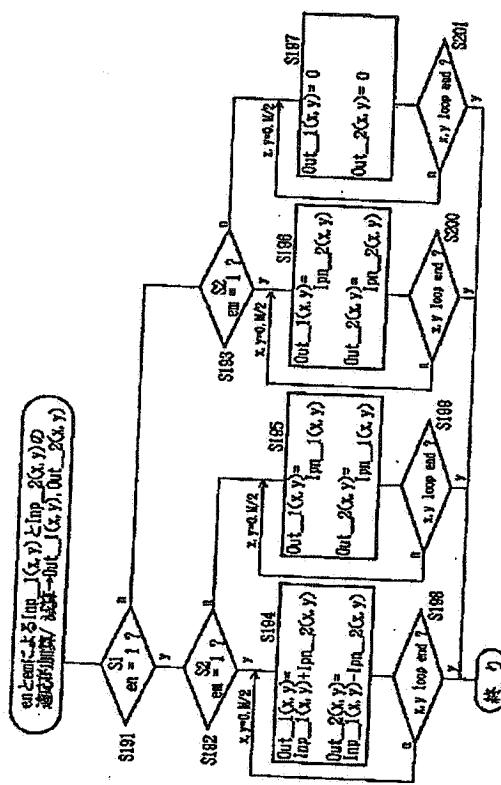


加算累積のフローチャート



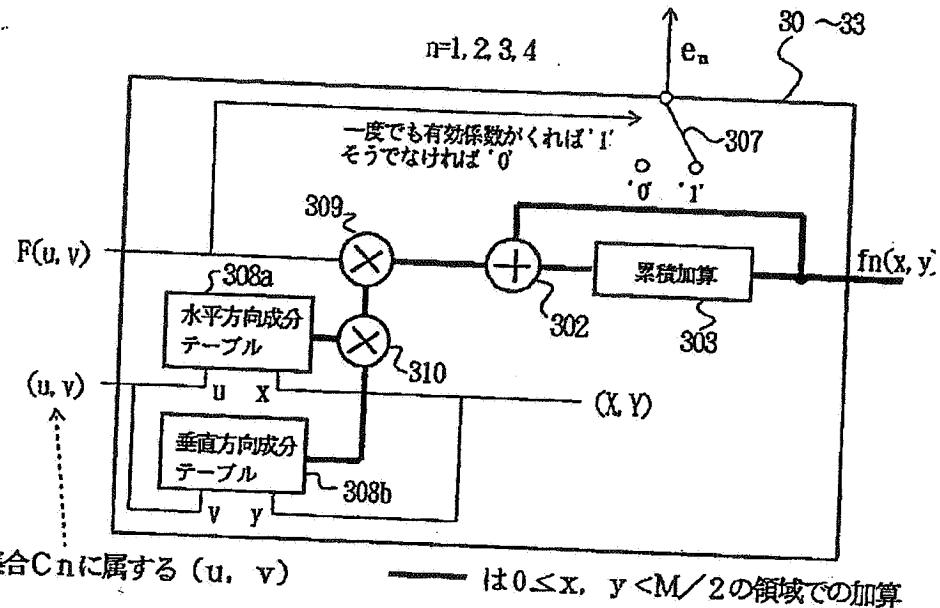
[图18]

適応的加算／減算のフロー・チャート



【図19】

加算累積部の構成例



[図20]

本発明の第五の実施例の原理ブロック図

